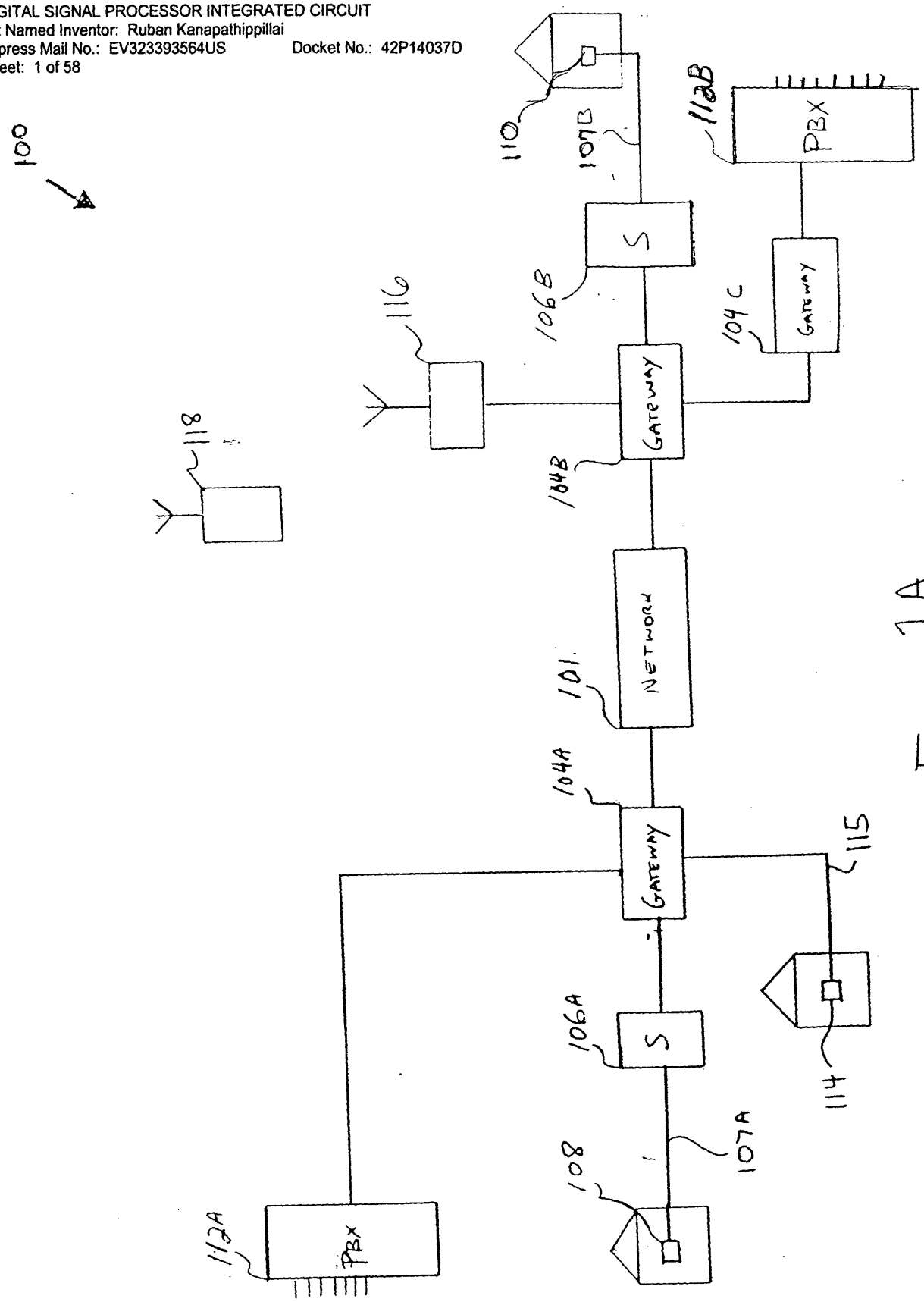

100
110
107B
112B
PBX
106B
S
104C
GATEWAY
116
118
104B
GATEWAY
101
NETWORK
104A
GATEWAY
115
106A
S
114
107A
108
112A
PBX

FIG. 1A

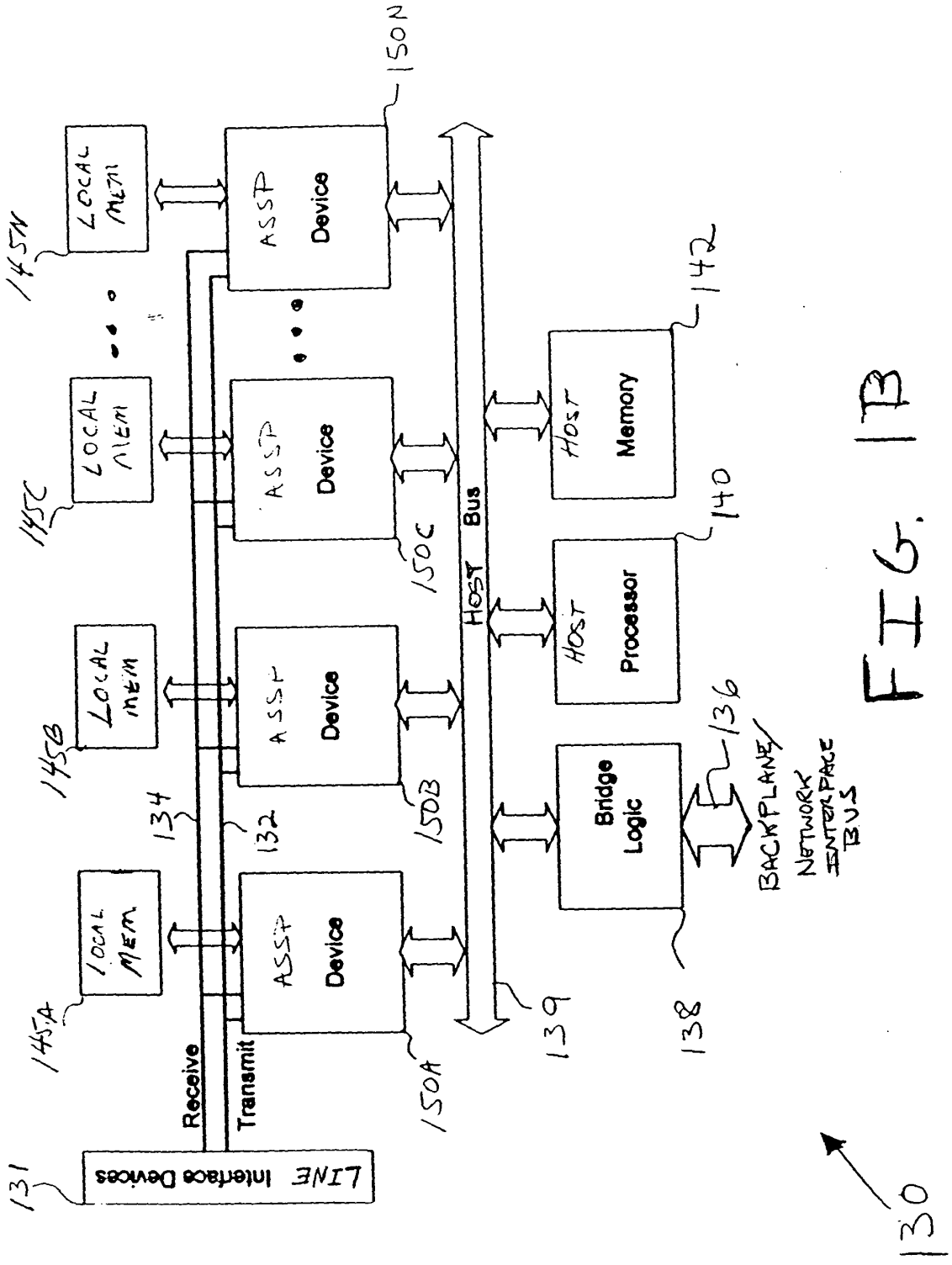

LINE Interface Devices
131
Receive
Transmit
134
132
Host Bus
ASSP Device
150A
150B
150C
150N
LOCAL MEM
145A
145B
145C
145N
Host Processor
140
Host Memory
142
Bridge Logic
136
138
139
BACKPLANE
NETWORK INTERFACE BUS
130

FIG. 1B

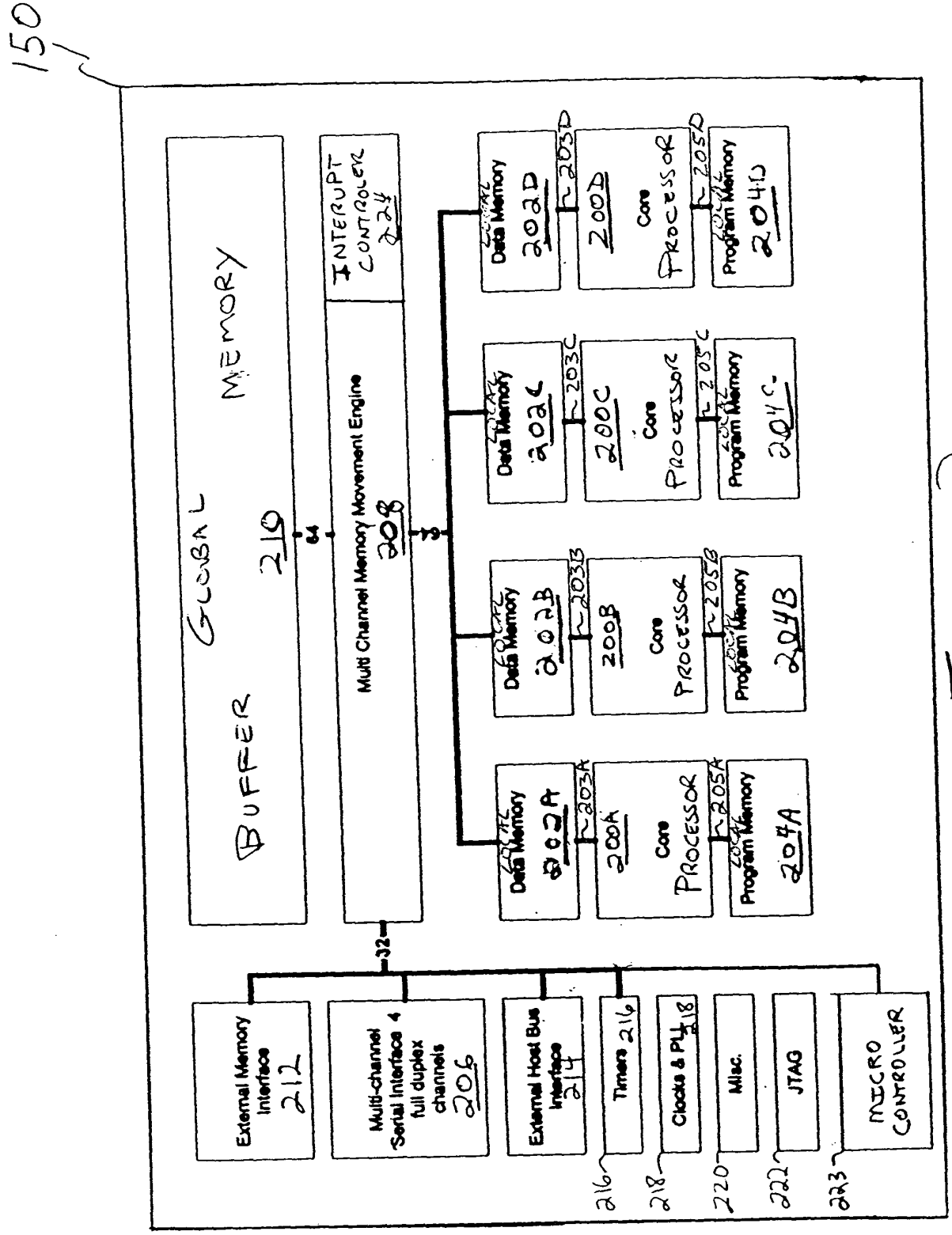

150
Global Memory Buffer 210
64
Interrupt Controller 224
Multi Channel Memory Movement Engine 208
32
Data Memory 202D
Core Processor 200D
Program Memory 204D
205D
203D
Data Memory 202C
Core Processor 200C
Program Memory 204C
205C
203C
Data Memory 202B
Core Processor 200B
Program Memory 204B
205B
203B
Data Memory 202A
Core Processor 200A
Program Memory 204A
205A
203A
External Memory Interface 212
Multi-channel Serial Interface 4 full duplex channels 206
External Host Bus Interface 214
Timers 216
Clocks & PLL 218
Misc.
JTAG
MICRO CONTROLLER
216
218
220
222
223

FIG. 2

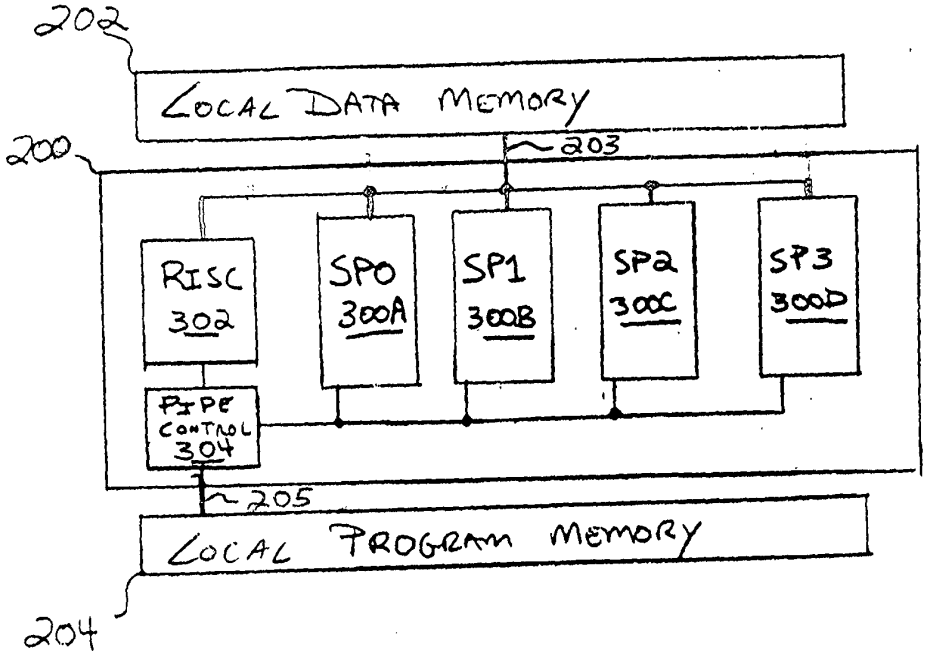


FIG. 3

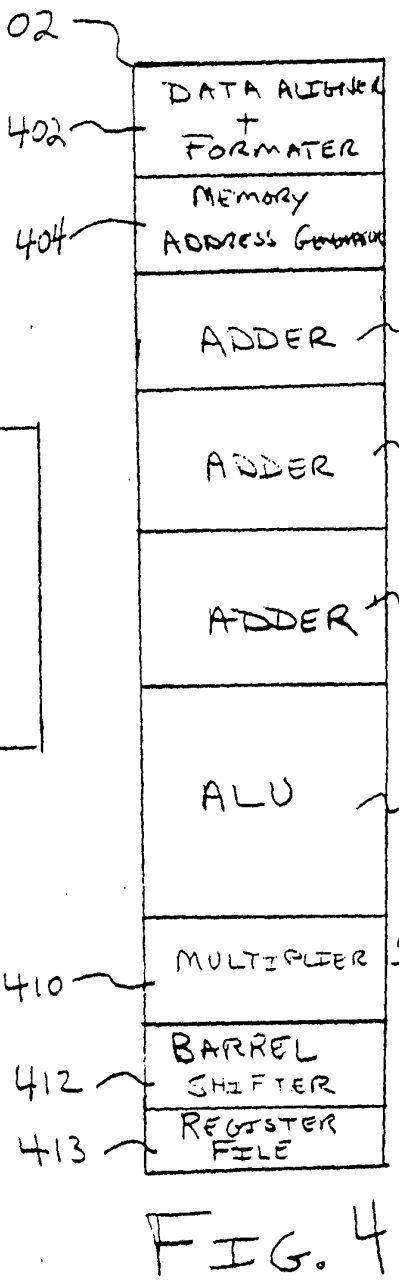


FIG. 4

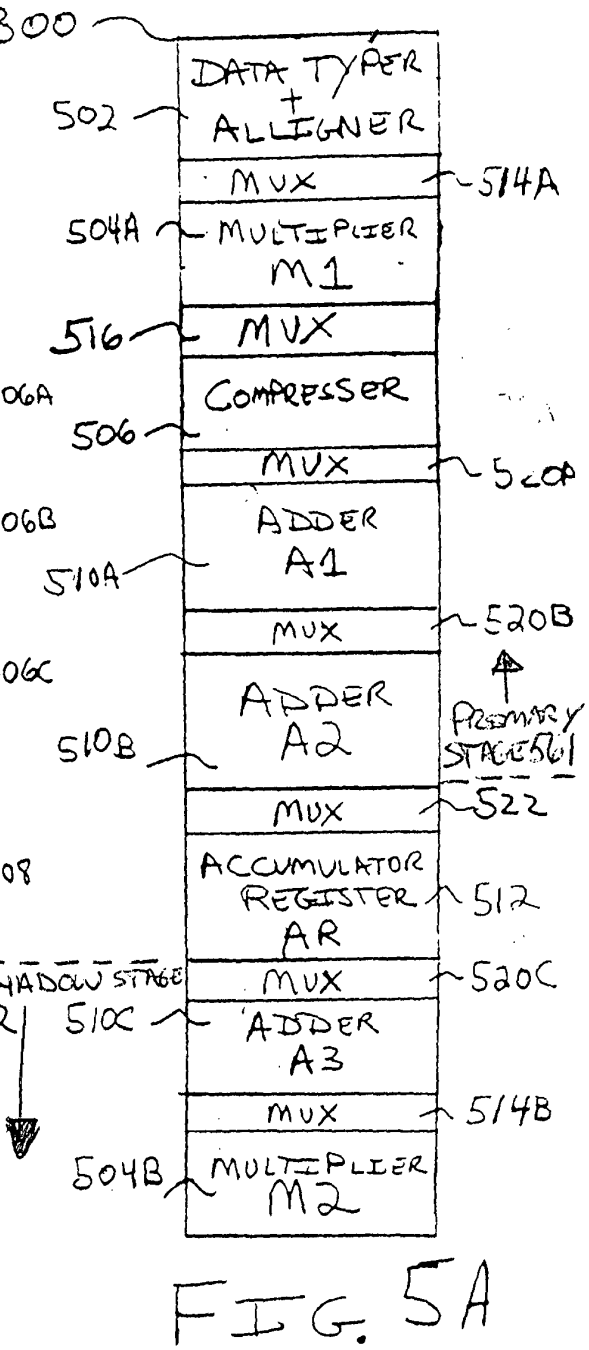


FIG. 5A

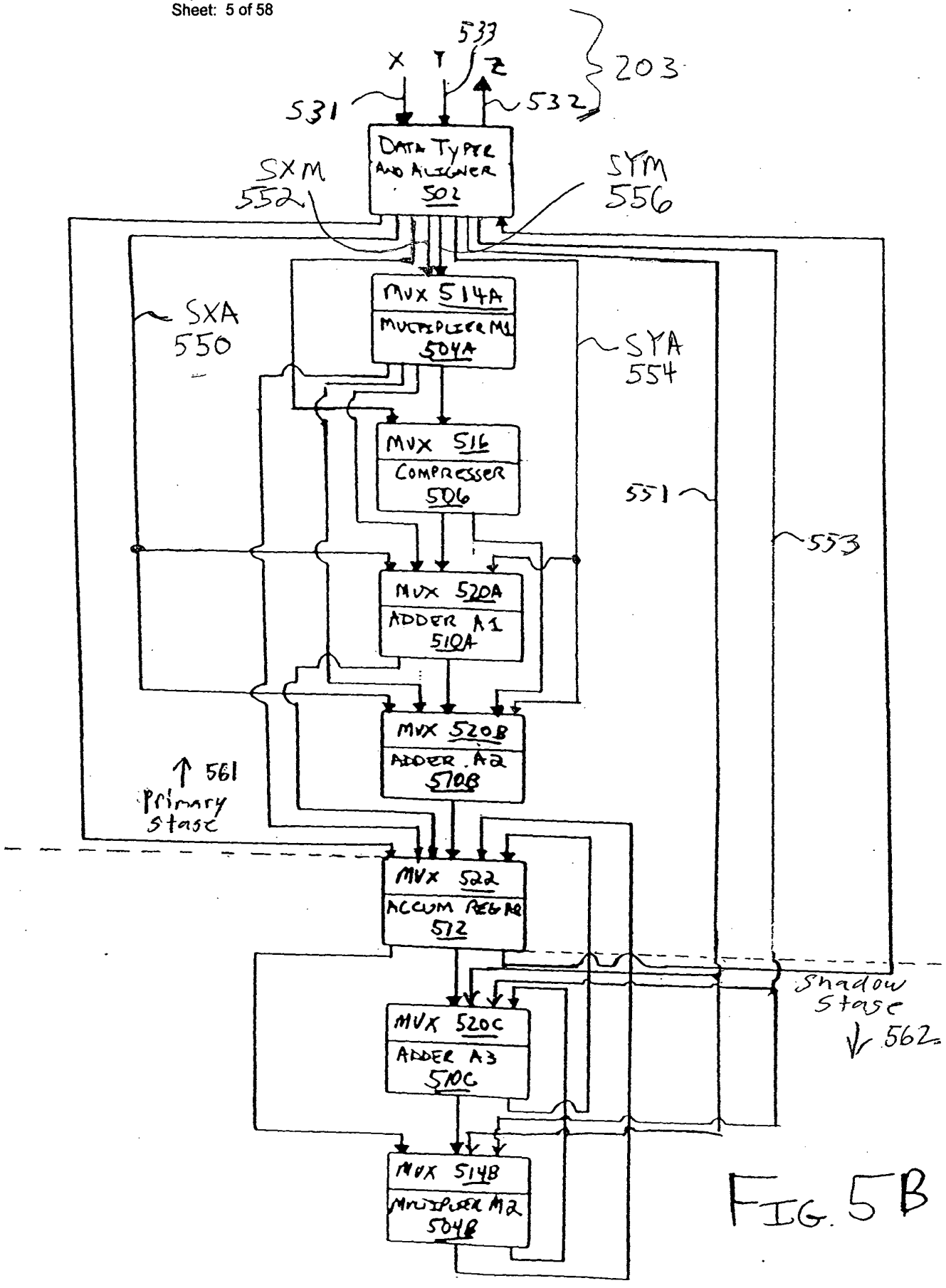

533
X
Y
Z
203
531
532
DATA TYPE AND ALIGNER 502
SXM 552
SYM 556
MUX 514A
MULTIPLIER M1 504A
SXA 550
SYA 554
MUX 516
COMPRESSER 506
551
553
MUX 520A
ADDER A1 510A
MUX 520B
ADDER A2 510B
561
Primary Stage
MUX 522
ACCUM REG M 512
Shadow Stage
562
MUX 520C
ADDER A3 510C
MUX 514B
MULTIPLIER M2 504B

FIG. 5B

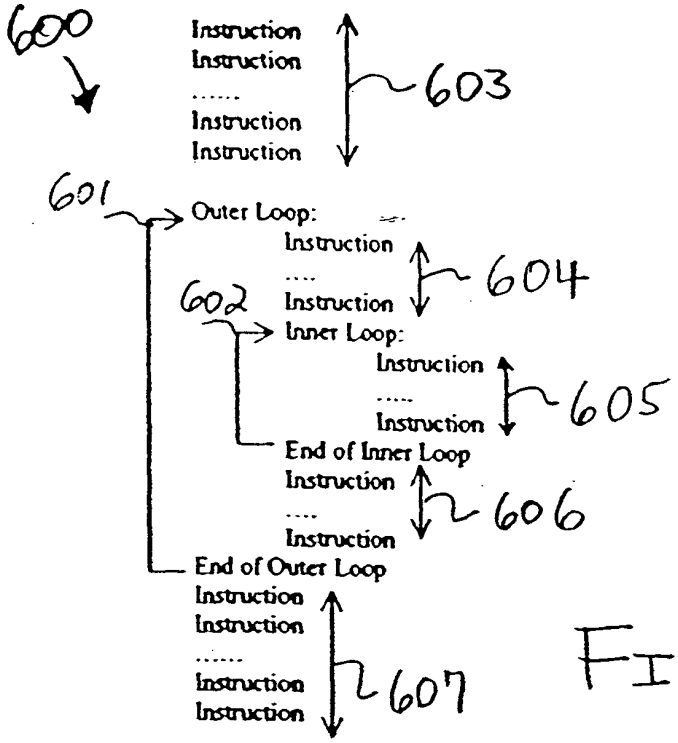


FIG. 6A

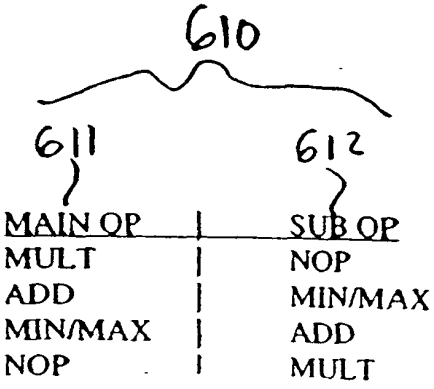


FIG. 6G

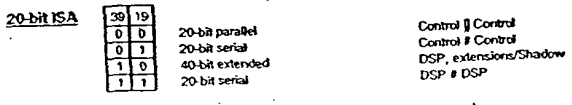


FIG. 6B

6-bit operand specifier:

A 6-bit specifier is used in DSP extended instructions to access memory and register operands.

5	4	3	2	1	0

M/R

0	0	ac-page	ereg
0	1	gpr: r0-r15	GPR
1	ptr : (r0) to (r15)	off	Mem[ptr[0-15]] || ptr[0-15] += offset1/offset2 Always postupdate

This allows access to data memory, ereg and GPR

- Bit 5 = 1: Use rX (X: 0-7) register to obtain effective memory address and post-modify the ptr field by one of two possible offsets specified in rX registers.
  dmem[ptr], ptr = ptr + offset1, if off = 0
  ptr = ptr + offset2, if off = 1
- Bit 5 = 0: Access ac-page or GPR

If Bit-4 is set to 0, then bits 3:0 control access to the general-purpose register file (r0-15) or to execution unit registers.

GPR
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15

GPR Intr page
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12_i
R13_i
R14_i
R15_i

ac-page
A0
A1
T
TR

ac intr page
A0_i
A1_i
T
TR

ereg-Shadow DSP
A0
A1
T
TR
SX1
SX1s
SX2
SX2s
SY1
SY1s
SY2
SY2s

FIG. 6C

For shadow DSP instructions, the 3-bit specifier for operands is defined as follows:

2	1	0	
0	0	0	A0
0	0	1	A1
0	1	0	T
0	1	1	TR
1	0	0	SX1
1	0	1	SX1s
1	1	0	SX2
1	1	1	SX2s

EREG1

2	1	0	
0	0	0	A0
0	0	1	A1
0	1	0	T
0	1	1	TR
1	0	0	SY1
1	0	1	SY1s
1	1	0	SY2
1	1	1	SY2s

EREG2

FIG. 6E

Only the shadow DSP instructions can see the above modified page of execution unit registers.

4-bit operand specifier:

Memory operands: (rX) specifies an access out of the data memory to the execution unit for the function that needs to be performed. The address for the access is specified in the rX register in the general register file that hold the 14-bit pointer (16K of addressing) to memory, 5-bit signed offset or a 3-bit unsigned offset that can post-modify the address. In addition each pointer is typed for efficient SIMD processing and includes a permute control for rearranging data elements of a vector on the fly. The "podi" core can deal with 4-element 16-bit real vectors or complex data directly. This ability to manipulate memory data directly reduces the instruction width greatly and allows efficient signal processing.

(rX): Memory Address Registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
type				cb	x	permute				off1: (0-7)			off0: (-16 to 15)					ptr: pointer													

FIG. 6D

5-bit operand specifier:

The 5-bit specifier includes the 4-bit specifier for general data operands and the special purpose registers. It is used in RISC instructions.

4	3	2	1	0
0	spr: s0-s15			
1	gpr: r0-r15			

	SPR	Intr page SPR intr page	
0	fu-ctl	fu-ctl_I	
1	a-type	a-type_I	
2	ps-ctl	ps-ctl	
3	t-type	t-type	
4	pl-ctl	pl-ctl	
5	cb-ctl	cb-ctl_I	
6	shuffle	shuffle	
7	io-ptr	io-ptr	
8	status	status_I	
9	loop-ctl	loop-ctl	
10	pcr	pcr	stack(8)
11	reserved	reserved	
12	reserved	reserved	
13	reserved	reserved	
14	reserved	reserved	

NOTE: All SPR registers are reset to all zeros at power on reset except for the PCR register.

FIG. 6F

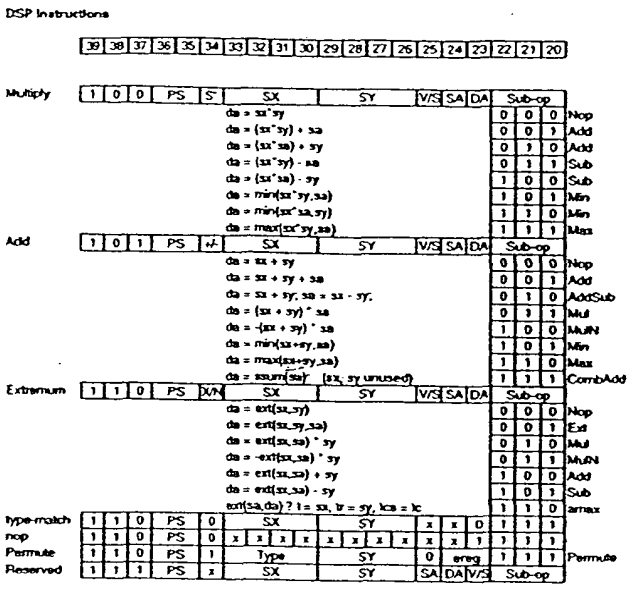

DSP Instructions
Multiply
Add
Extremum
type-match
nop
Permute
Reserved
Nop
Add
Sub
Min
Max
AddSub
Mul
MulN
CombAdd
Ext
amax
Permute

FIG. 6H

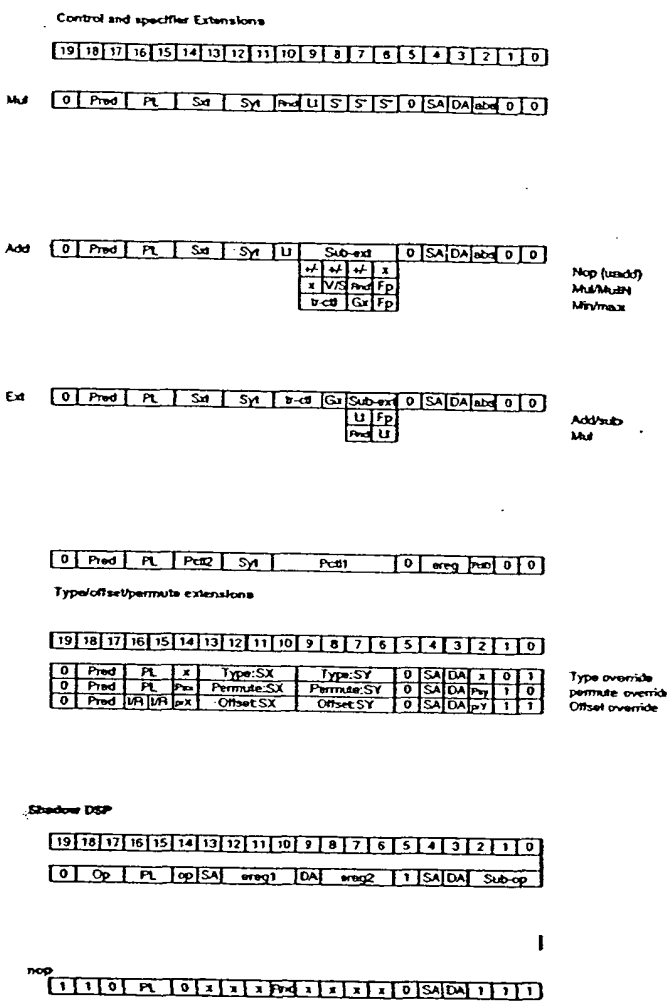

Control and specifier Extensions
Mul
Add
Ext
Nop (usadd)
Mul/MulN
Min/max
Add/sub
Mul
Type/offset/permute extensions
Type override
permute overrid
Offset override
Shadow DSP
nop

FIG. 6I

Control Instructions

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
add_sub	L	Pred	0	0	0	RX					RY				RZ				+/-	0
max,min	L	Pred	0	0	0	RX					RY				RZ				U/S	1
Shift	L	Pred	0	0	1	RX					U14				RZ				U1	R/L
Logic	L	Pred	0	1	0	RX					RY				RZ				&	| &!
Mux	L	Pred	0	1	1	RX					RY				RZ				Pd	0
mov	L	Pred	0	1	1	RX					DZ				Pdt	Dzt	0	0	0	1
addi	L	Pred	0	1	1	SI4					DZ				x	x	1	0	0	1
mov2erg	L	Pred	0	1	1	RX				unit	ereg				gd	W/Eg	s:s	1	0	1
Ldm	L	Pred	0	1	1	RX					DZ1				DZ2				1	1
Set4bits	L	Pred	1	0	0	U14:POS					RZ				Pzt	U14				0
Set2bits	L	Pred	1	0	0	U14:POS					RZ				Pzt	U12		0	0	1
Setbit	L	Pred	1	0	0	U14:POS					RZ				Pzt	U11	U11	1	0	1
Movi	L	Pred	1	0	0	SI8									RZ				1	1
Jmp	L	Pred	1	0	1	SI9												0	Pred	0	0
Call	L	Pred	1	0	1	SI9												1	Pred	0	0
Loop	L	Pred	1	0	1	U15 : Lcount					U15: Lsize							U12: Ls		0	1
Jmpi	L	Pred	1	0	1	RX	x	x	x	x	x	0	Pred		1	0				
Calli	L	Pred	1	0	1	RX	x	x	x	x	x	1	Pred		1	0				
Loopi	L	Pred	1	0	1	RX	x	U15: Lsize							U12: Ls					1	1
Test	L	Pred	1	1	0	RX					RY				PZ				=,<,>	0
Testbit	L	Pred	1	1	0	RX					U15				PZ				B	0	1
Andp, orp	L	Pred	1	1	0	Pa			Pb			Pc			PZ				&|	1	1
Load	L	Pred	1	1	1	MX					RZ				Ext				0	0	0
Store	L	Pred	1	1	1	MZ					RX				Ext				1	0	0
eLoad	L	Pred	1	1	1	MX					RZ				1	1	1	0	0	0
eStore	L	Pred	1	1	1	MZ					RX				1	1	1	1	0	0
Extended	L	Pred	1	1	1	Bits 27:16													1	0
Logic2	L	Pred	1	1	1	RX					RY/RZ				Rzt	Ryt	U4	&|	0	1
mov-erg	L	Pred	1	1	1	unit	ereg				RZ				gd	SR		0	1	1
Crb	L	Pred	1	1	1	RX					RZ				s/m	0	0	1	1	1
Parity	L	Pred	1	1	1	RX					PZ			O/E	0	1	0	1	1	1
Stm	L	Pred	1	1	1	MZ					RX				1	1	0	1	1	1
Abs	L	Pred	1	1	1	RX					RZ				0	0	1	1	1	1
Neg	L	Pred	1	1	1	RX					RZ				0	1	1	1	1	1
Div-step	L	Pred	1	1	1	RX					RZ				1	0	1	1	1	1
Reserved	L	Pred	1	1	1	RX					RY/RZ				0	0	0	1	1	1
Reserved	L	Pred	1	1	1	RX					RY/RZ				0	0	0	1	1	1
Return	L	Pred	1	1	1	Pred	I-cll				0	1	0	1	1	1	1	1	1	1
Zero ac	L	Pred	1	1	1	ac #					1	1	0	1	1	1	1	1	1	1
eSync	L	Pred	1	1	1	RZ					0	1	1	1	1	1	1	1	1	1
Swi	L	Pred	1	1	1	U13				0	1	1	1	1	1	1	1	1	1	1
Nop	L	Pred	1	1	1	U13				1	1	1	1	1	1	1	1	1	1	1

<Sl41, Bits9-6> = UI5 (Shift Amount)

Bit 5: 0=one ereg, 1=broadcast all four; Bit4: 0=16-bit, 1=32-bit

<Bit3, Bits13-10> = UI5 :POS

FIG. 6J

Extended Control

	Bits 13:2 of upper half (39:20)										19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Insert/Extract	RX				RZ						0	0	0	0	x	x	0	Rzt	Rzt	I/E	R/I	R/I	Offset:UI5					Length:UI5			0
																				x	RY				RY				x	
Insert	UI4: length				RZ						0	0	0	1	0	x	x	0	Rzt	UI5: Position					Imm10					
Shift	RX				RZ						0	0	0	0	0	rxh	rzl	0	rzh	rzl	D	U/S	1	Shift: UI5				A/L	LI	R/L	0	Fill	1
																							0	ryh	RY							
Rotate	RX				RZ						0	0	0	0	0	x	x	0	x	x	x	x	1	Shift: UI5				1	x	R/L	1	x	1
																							0	ryh	RY							
jmp, call	UI7							JIC	0	0	1	0	0	Pred		0	R	UI15														
dloop	UI4: outer LC				UI4: inner LC				0	0	1	1	0	x	exit	0	UI1	UI4: outer Lsize				UI4: inner Lsize			uz: 0-Ls		UI4: inner Lstart			0
dloopi	RX				RY				0	0	1	1	0	x	exit	0	x	UI4: outer Lsize				UI4: inner Lsize			uz: 0-Ls		UI4: inner Lstart			1
mult	RX				RY				0	1	0	0	0	x	Imx	0	0	rxh	mcl	ryh	+/-	=/+	RZ			vl	rzh	rzl	s/u	s/u	0
add/sub	RX				RY				0	1	0	0	0	x	x	0	0	rxh	rxl	ryh	ryl	+/-	RZ			LI	rzh	rzl	x	x	1
Reserved	RX				RY				0	1	0	0	0	x	x	0	0	rxh	rxl	ryh	ryl	+/-	RZ			LI	rzh	rzl	x	x	1
logicp	PX		D	PZ					0	1	0	0	0	x	x	0	1	T/F	T/F	T/F	&|	&|	PY			PY			x	1
Testl	RX		D	PZ					0	1	0	1	0	=,>,<		0	Imm16														
Movi	HWI	Fill	RZ						0	1	1	0	0	x	x	0	Imm16														
loadi	Type		RZ						0	1	1	1	0	x	x	0	0	0	Imm14												
storei	Type		RX						0	1	1	1	0	x	x	0	0	1	Imm14												
loadi	RX		RZ						0	1	1	1	0	x	x	0	1	Rzt	0	Type	SI10										
storei	MZ		RX						0	1	1	1	0	x	x	0	1	Rzt	1	Type	SI10										
Add/subi	RX		RZ						1	0	+/-	0	0	L1	s/u	0	Imm16														
min/maxi	RX		RZ						1	0	U/S	1	0	x	x	0	Imm16														
andi,ori	RX		RZ						1	1	&|	H/L	0	x	x	0	Imm16														

Fill: Sign/Zero

RI = PC relative
Bit 15 is continuation of inner LC

andp, orp, andorp, orandp: pz = (px relop py) relop pv

FIG. 6K

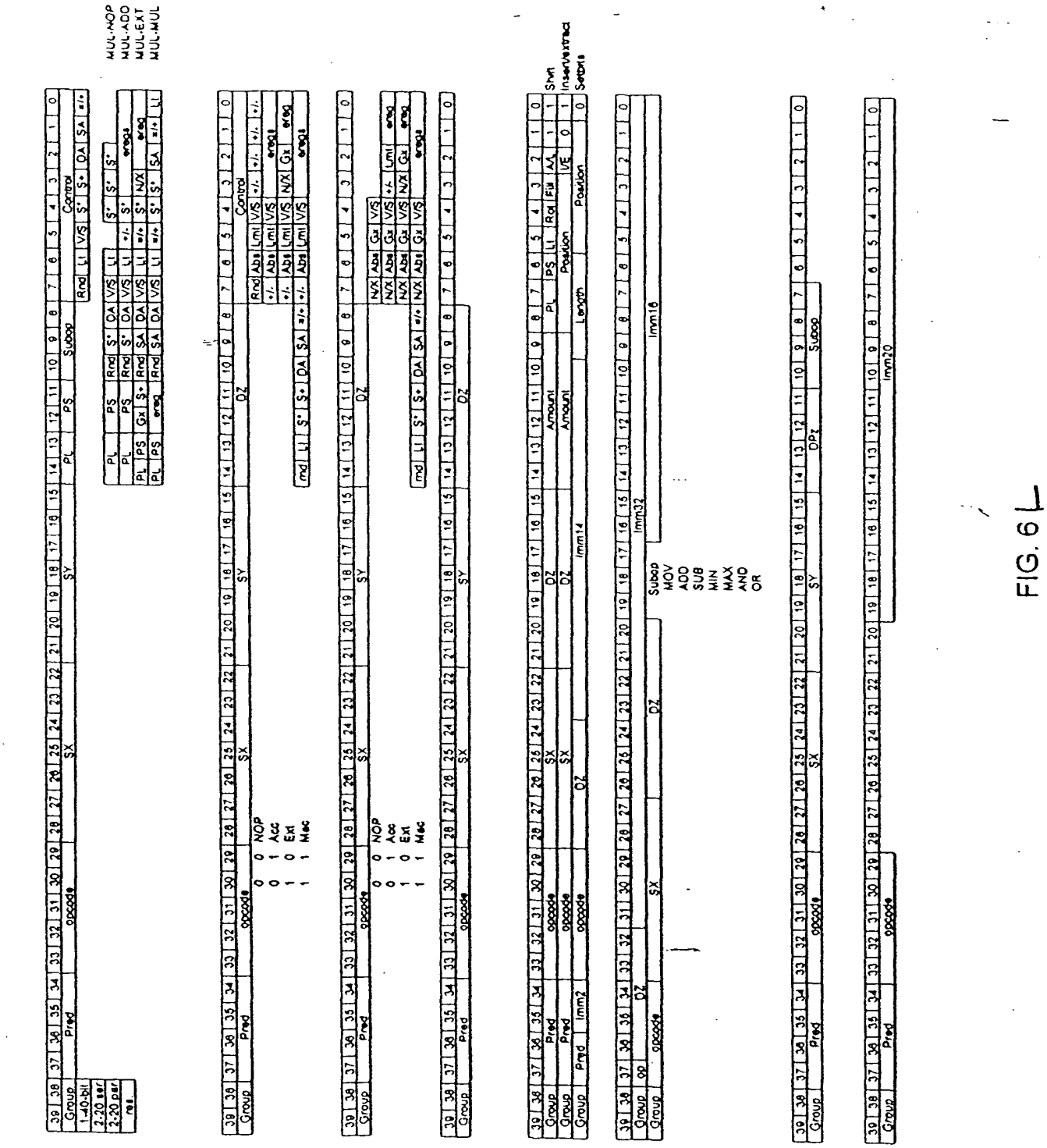

MAC:
ARITH:
EXT:
LOGIC:
SHIFT:
Immediate:
Test:
Branch:
Misc:
MUL.NOP
MUL.ADD
MUL.EXT
MUL.MUL
NOP
Acc
Ext
Mac
MOV
ADD
SUB
MIN
MAX
AND
OR

FIG. 6L

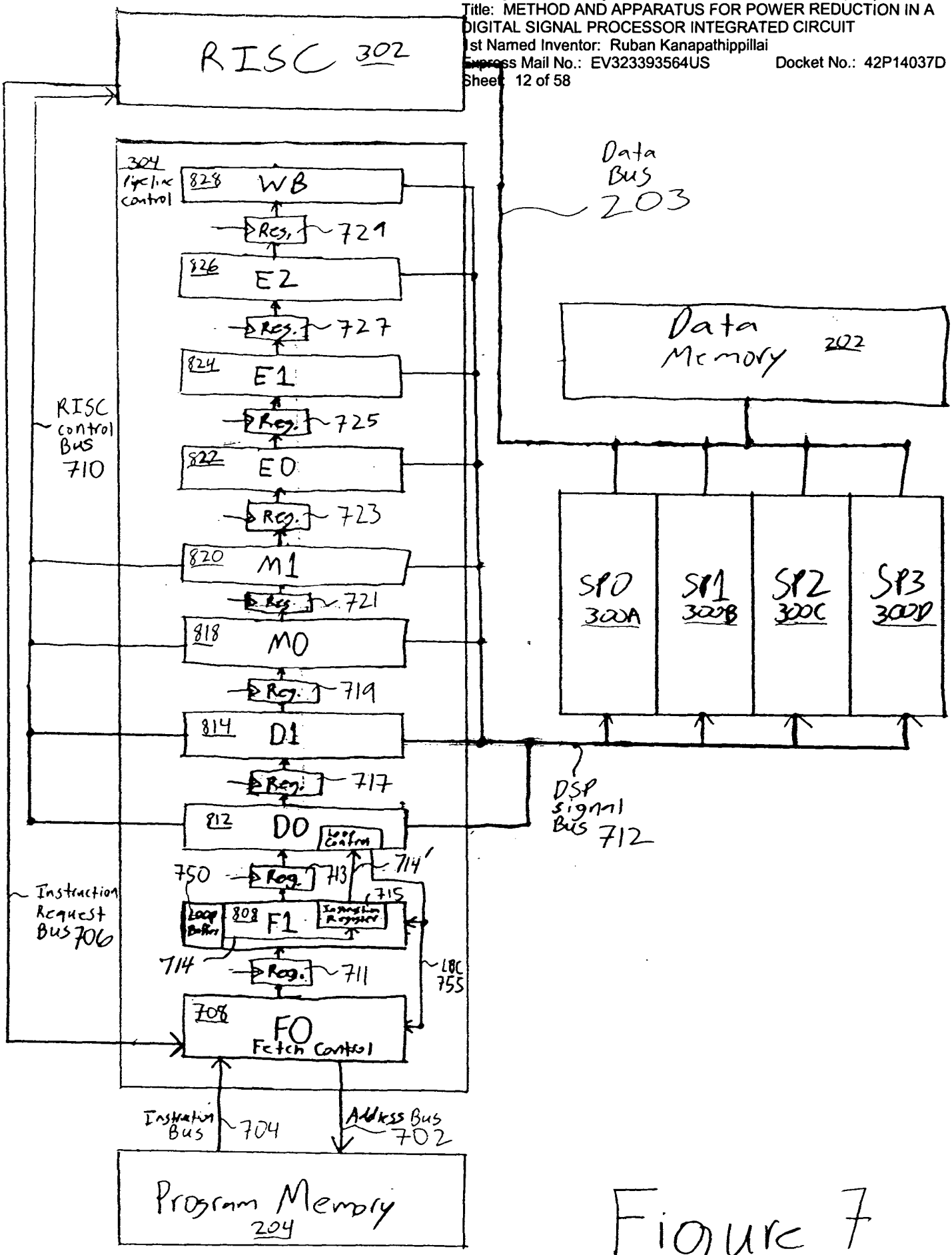

RISC 302
304 Cyclic control
828 WB
Reg. 721
826 E2
Reg. 727
824 E1
Reg. 725
822 E0
Reg. 723
820 M1
Reg. 721
818 M0
Reg. 719
814 D1
Reg. 717
812 D0
Write Confirm
714'
750
Reg. 713
715
Loop Buffer
808 F1
Instruction Register
714
Reg. 711
LBC 755
708 F0 Fetch Control
Instruction Bus 704
Address Bus 702
Program Memory 204
RISC control Bus 710
Instruction Request Bus 706
Data Bus 203
Data Memory 202
SP0 300A
SP1 300B
SP2 300C
SP3 300D
DSP signal Bus 712

Figure 7

Pipeline Controller 304

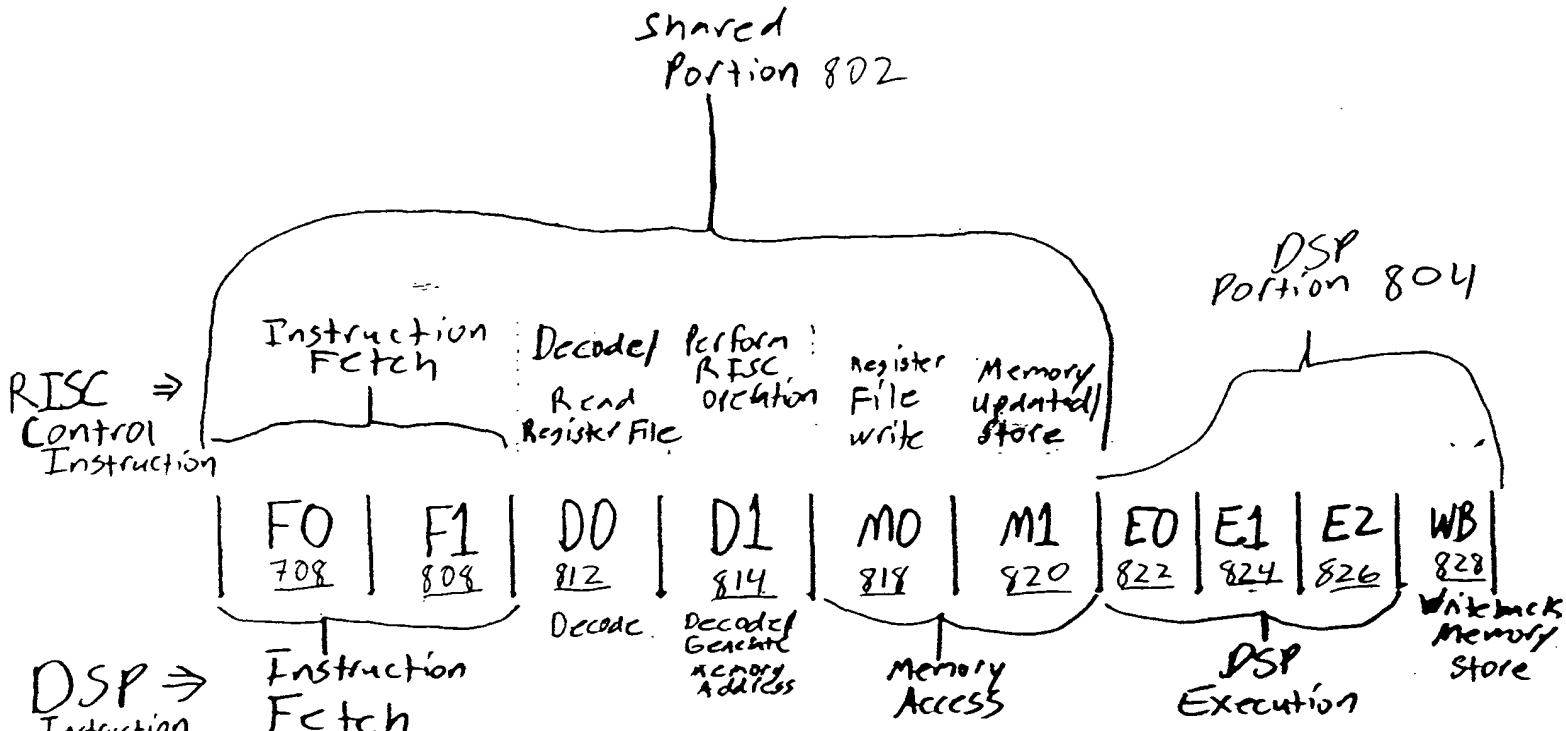


Figure 8A

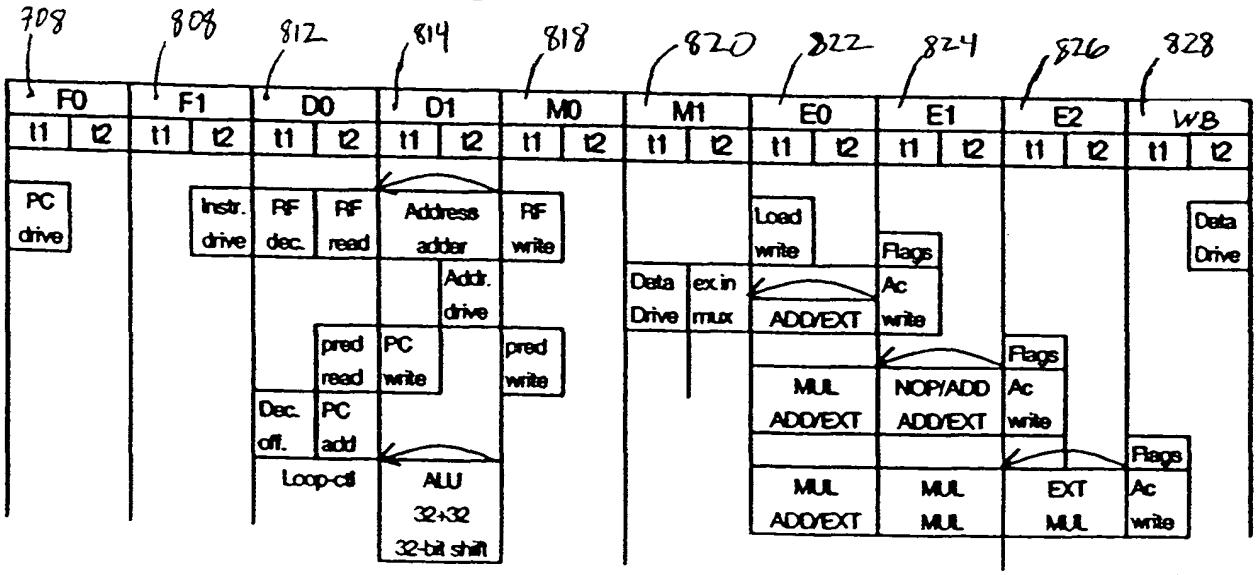


Figure 8B

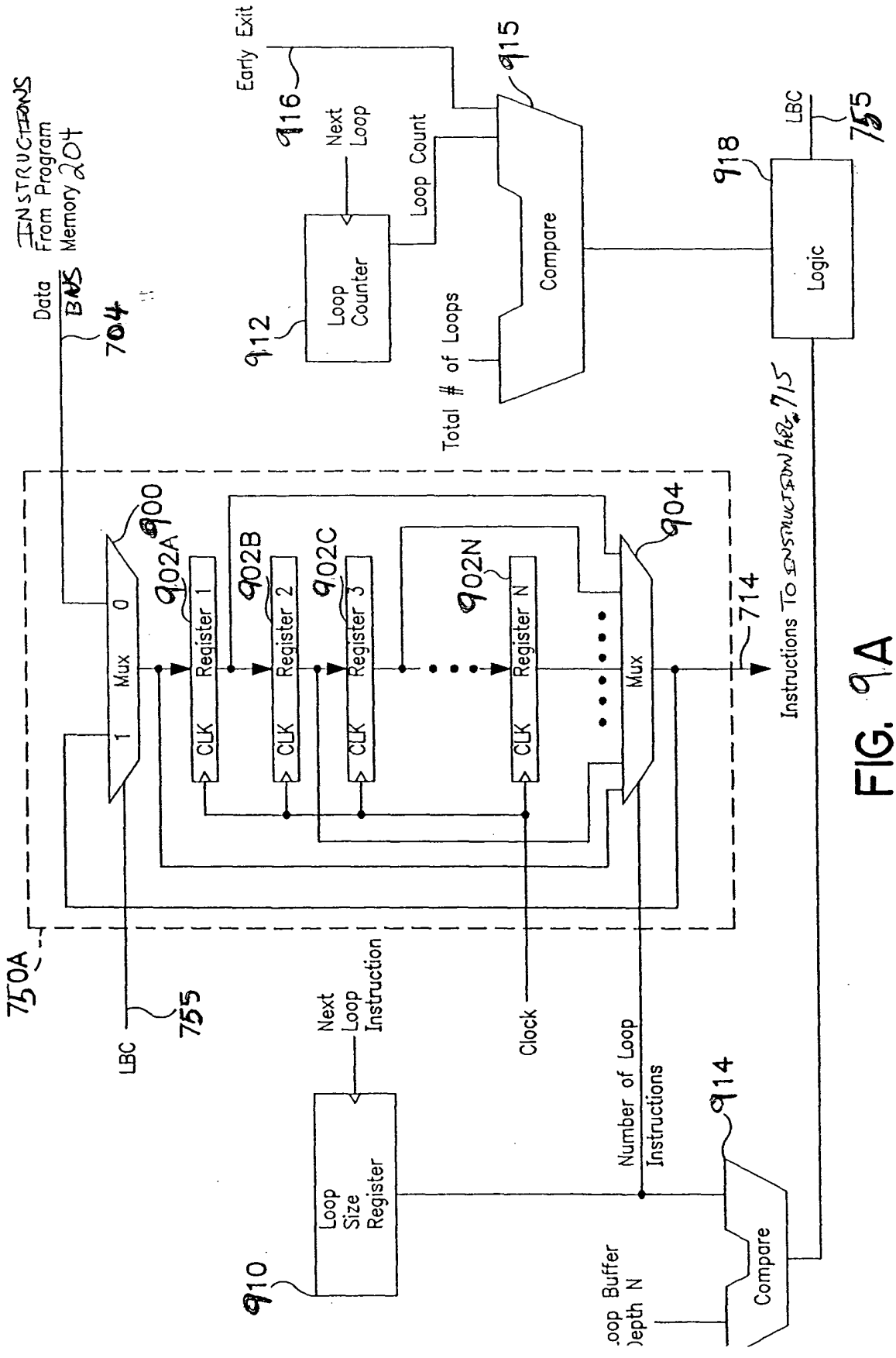

INSTRUCTIONS From Program Memory 204
Data BUS
704
Early Exit
916
Next Loop
Loop Count
915
Loop Counter
912
Compare
Total # of Loops
918
Logic
LBC
755
Instructions To Instruction Reg. 715
900
902A
902B
902C
902N
Register 1
Register 2
Register 3
Register N
CLK
Mux
0
1
904
714
750A
LBC
755
Next Loop Instruction
Clock
Number of Loop Instructions
914
Loop Size Register
910
Loop Buffer Depth N
Compare

FIG. 9A

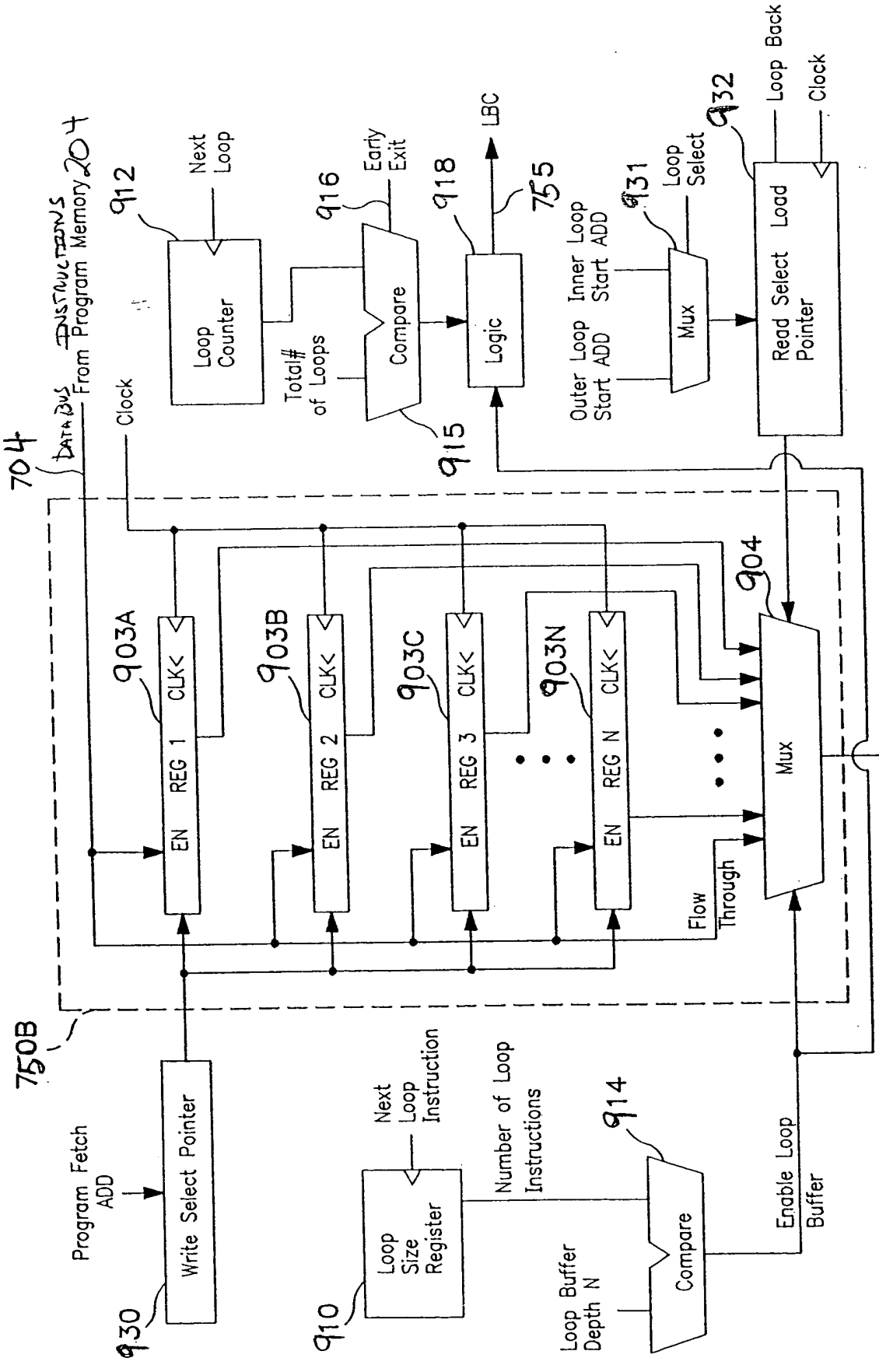

704
Data Bus
From Program Memory 204
Instructions
Clock
912
Loop Counter
Next Loop
916
Compare
Early Exit
915
Total # of Loops
918
Logic
LBC
755
931
Mux
Loop Select
Outer Loop Start ADD
Inner Loop Start ADD
932
Read Select Load Pointer
Loop Back
Clock
903A
903B
903C
903N
EN REG 1 CLK<
EN REG 2 CLK<
EN REG 3 CLK<
EN REG N CLK<
904
Mux
714
Instructions
TO INSTRUCTION REGISTER 715
Flow Through
750B
Program Fetch ADD
930
Write Select Pointer
910
Loop Size Register
Next Loop Instruction
Number of Loop Instructions
914
Compare
Loop Buffer Depth N
Enable Loop Buffer

FIG. 9B

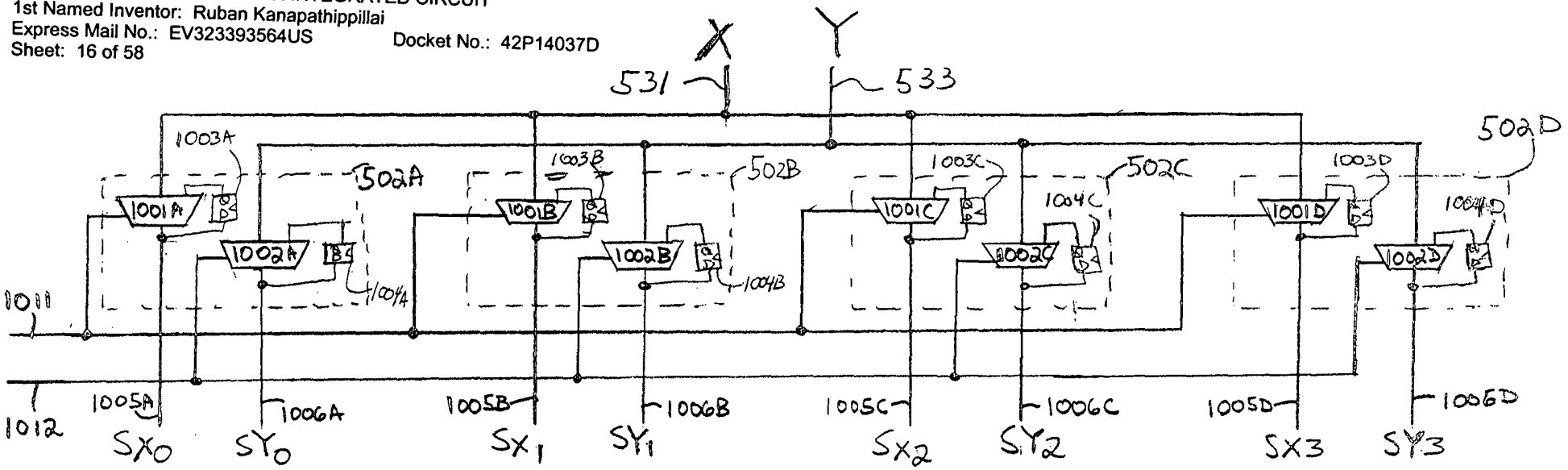

X
Y
531
533
502A
502B
502C
502D
1003A
1003B
1003C
1003D
1001A
1001B
1001C
1001D
1002A
1002B
1002C
1002D
1004A
1004B
1004C
1004D
1011
1012
1005A
1005B
1005C
1005D
1006A
1006B
1006C
1006D
SX0
SY0
SX1
SY1
SX2
SY2
SX3
SY3

FIG. 10

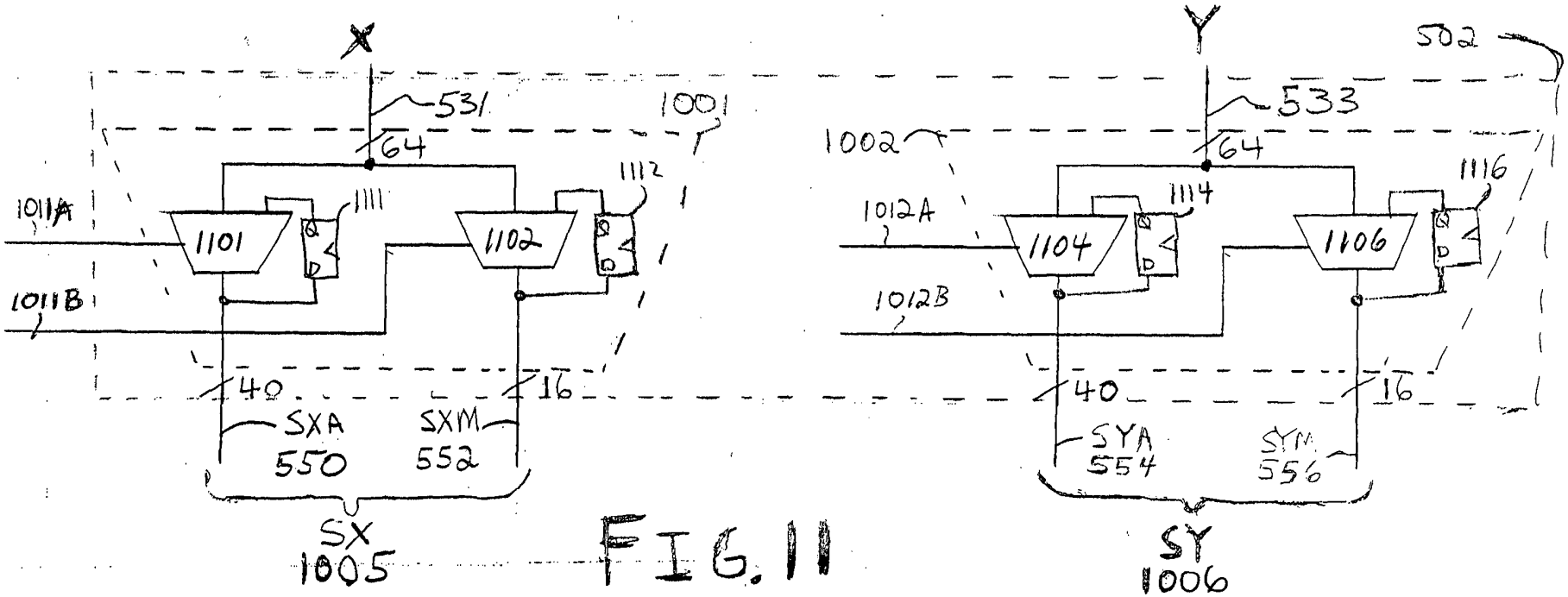
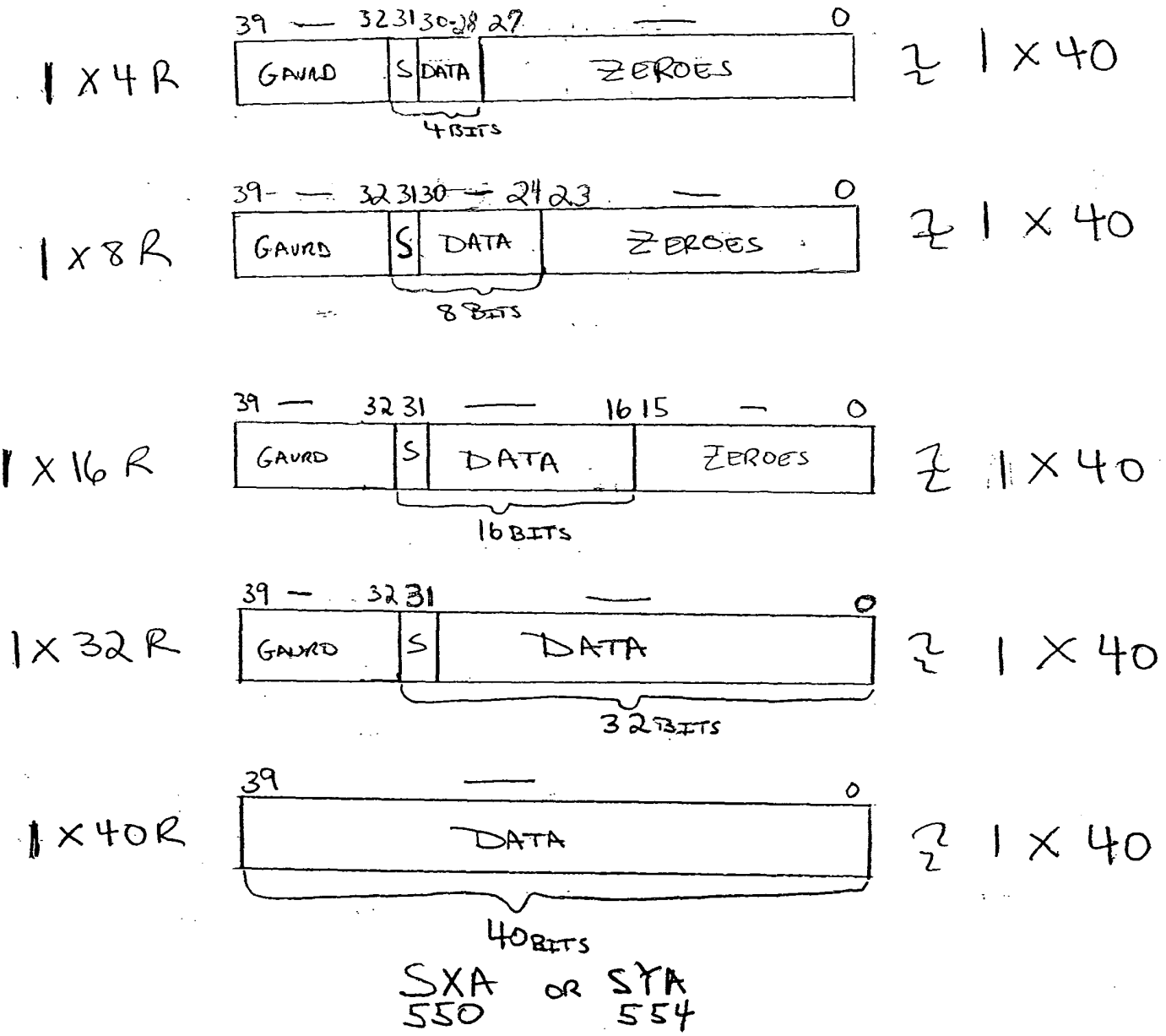

X
Y
531
533
1001
1002
502
64
64
1111
1112
1114
1116
1101
1102
1104
1106
1011A
1011B
1012A
1012B
40
16
40
16
SXA
550
SXM
552
SYA
554
SYM
556
SX
1005
SY
1006

FIG. 11

DATA TYPE

SP CONFIGURATION



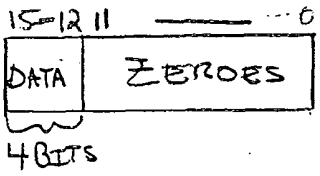
SXA 550 OR SYA 554

FIG. 12A

DATA TYPE

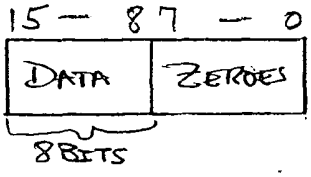
SP CONFIGURATION

1 X 4 R



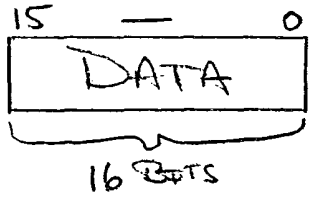
→ 1 X 16

1 X 8 R



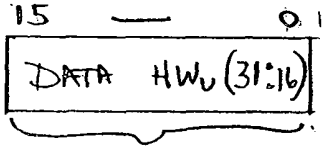
→ 1 X 16

1 X 16 R



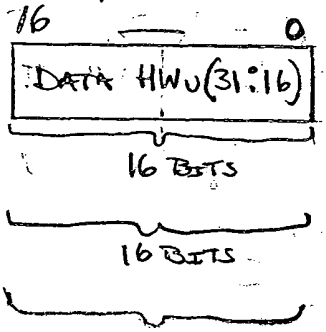
→ 1 X 16

1 X 32 R



→ 1 X 16

1 X 40 R



→ 1 X 16

SXM 552A - 552B
OR
SYM 556A - 556B

FIG. 12B

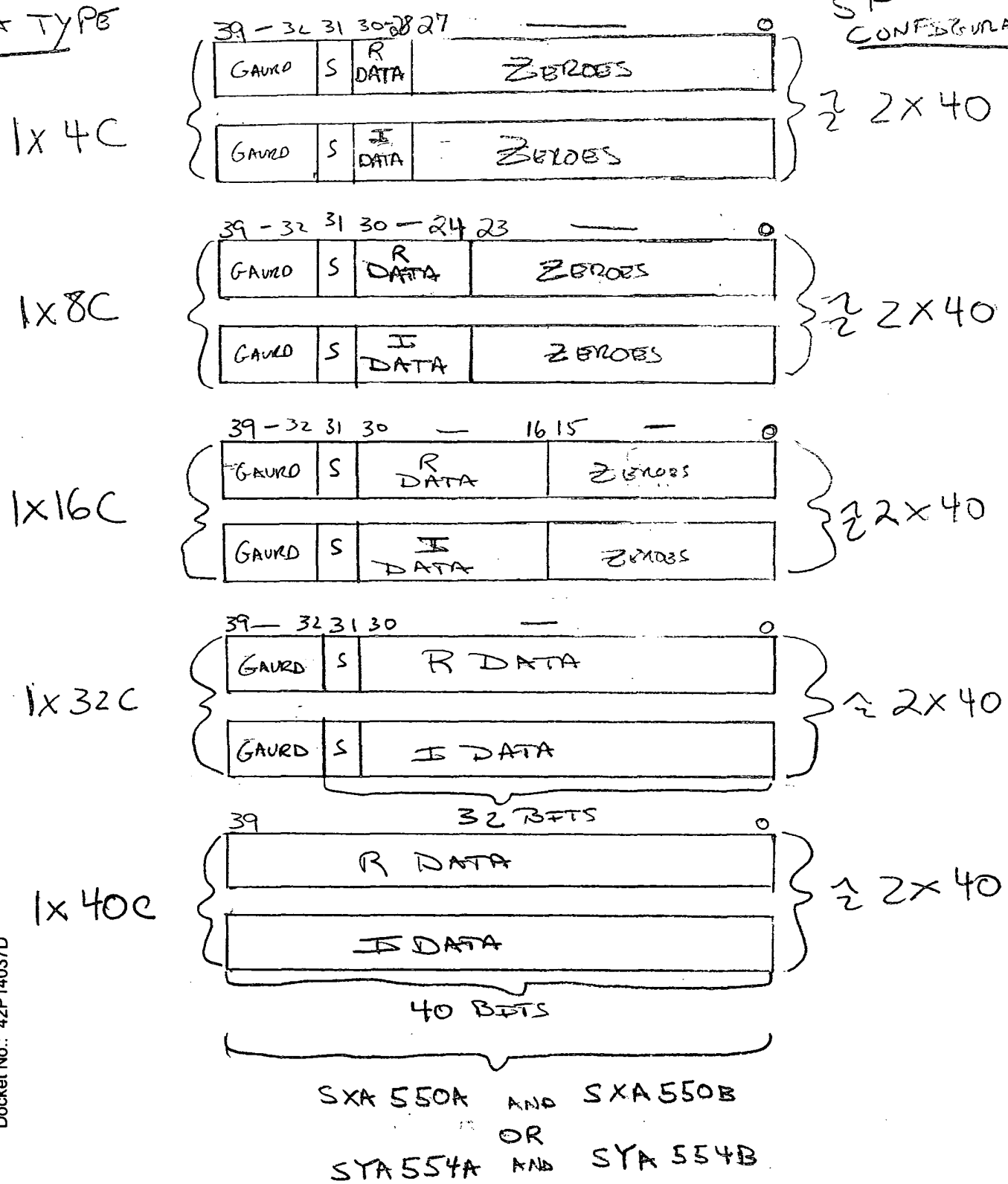


FIG. 12C

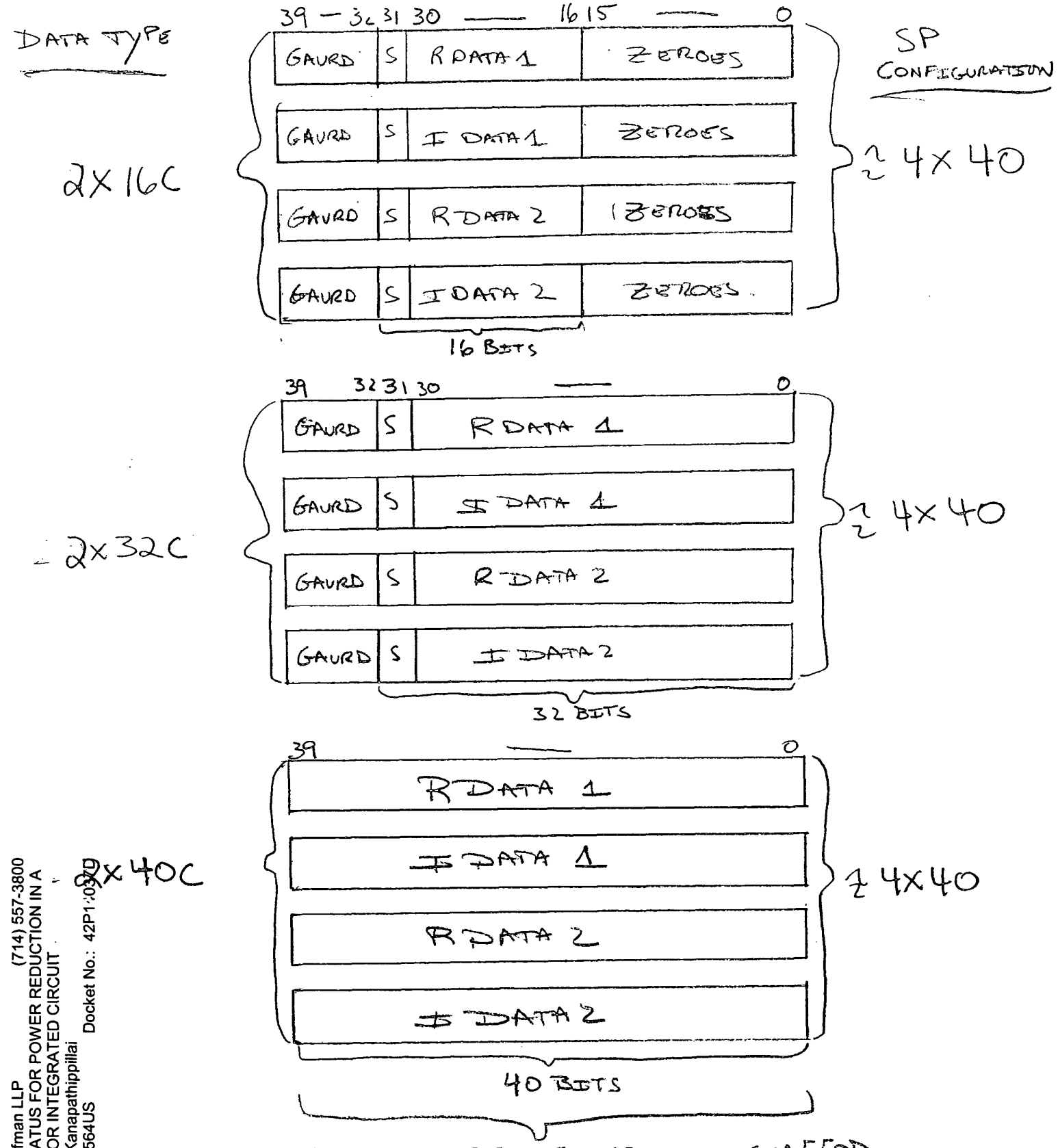
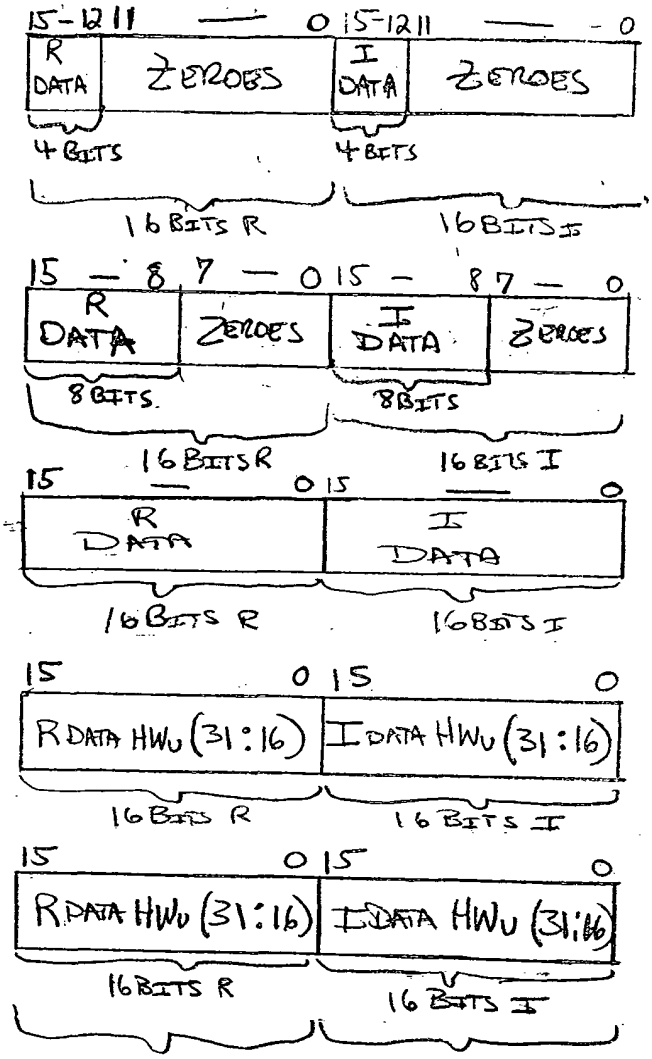


FIG. 12D

PATA TYPES

SP CONFIGURATION



½ 2X16

½ 2X16

½ 2X16

½ 2X16

½ 2X16

SXM552A AND SXM552B

OR

SYM556A AND SYM556B

FIG. 12E

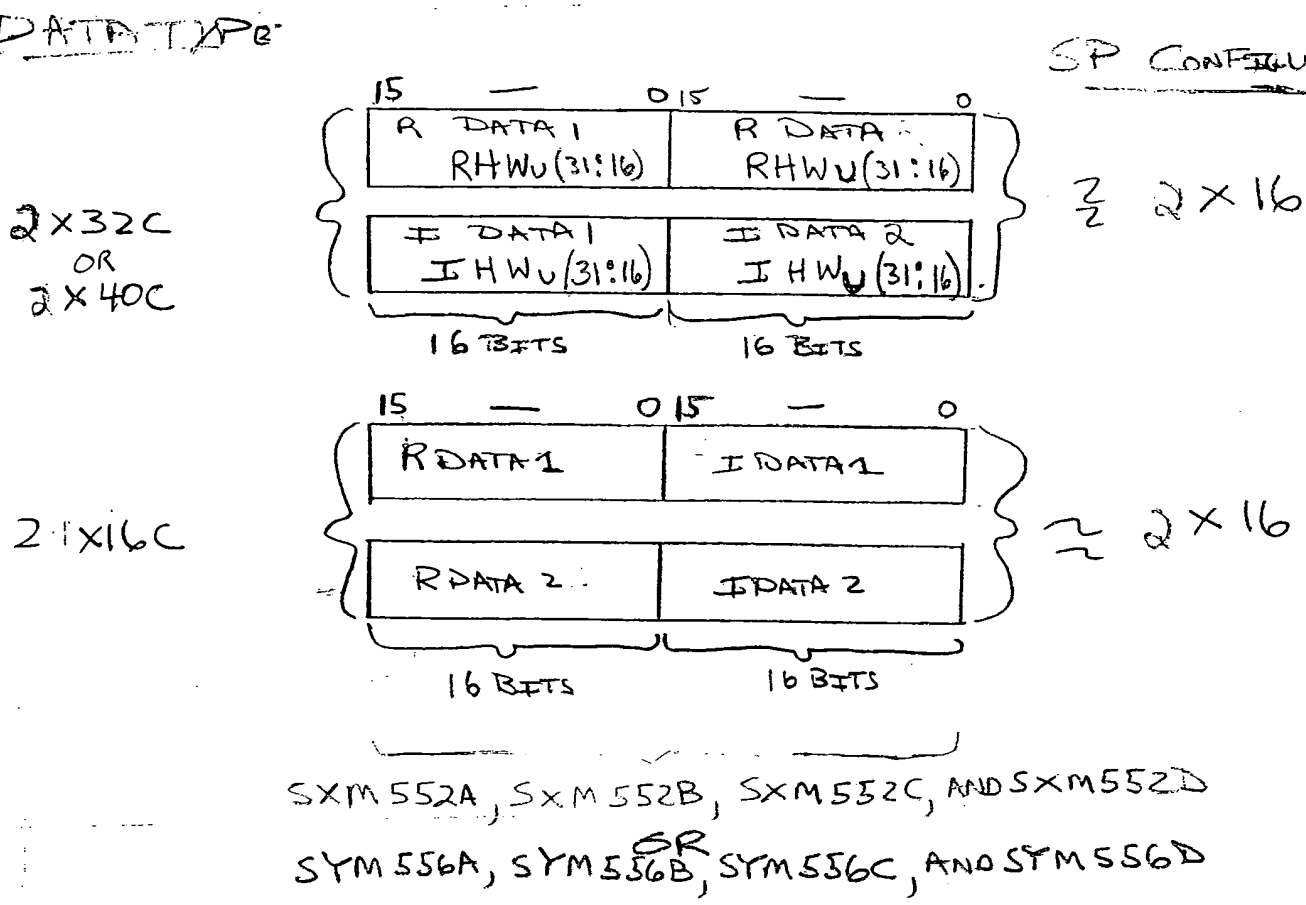


FIG. 12F

Operand 1 Data Type:	$N_1 \times S_1$ R
Operand 2 Data Type:	$N_2 \times S_2$ R
Type Matching R:	Max (N_1 or N_2) x Max (S_1 or S_2) R

Fig. 13A

Operand 1 Data Type:	$N_1 \times S_1$ C
Operand 2 Data Type:	$N_2 \times S_2$ C
Type Matching C:	Max (N_1 or N_2) x Max (S_1 or S_2) C

Fig. 13B

Operand 1 Data Type:	$N_1 \times S_1$ R
Operand 2 Data Type:	$N_2 \times S_2$ C
Type Matching R+C:	Max (N_1 or N_2) x Max (S_1 or S_2) C

Fig. 13C

	1x16 real	2x16 real	1x16 cmpx	4x16 real	2x16 cmpx	1x32 real	2x32 real	1x32 cmpx	4x32 real	2x32 cmpx	1x40 real	2x40 real	1x40 cmpx	4x40 real	2x40 cmpx
1x16 real	1 unit	2 unit	2 unit	4 unit	4 unit	2 unit	4 unit	4 unit							
2x16 real	2 unit	2 unit													
1x16 cmpx	2 unit		4 unit												
4x16 real	4 unit			4 unit											
2x16 cmpx	4 unit														
1x32 real	2 unit														
2x32 real	4 unit														
1x32 cmpx	4 unit														
4x32 real															
2x32 cmpx															
1x40 real															
2x40 real															
1x40 cmpx															
4x40 real															
2x40 cmpx															

FIG. 14

	1x16 real	2x16 real	1x16 cmpx	4x16 real	2x16 cmpx	1x32 real	2x32 real	1x32 cmpx	4x32 real	2x32 cmpx	1x40 real	2x40 real	1x40 cmpx	4x40 real	2x40 cmpx
1x16 real	1 unit	2 unit		4 unit		1 unit	4 unit		4 unit		1 unit	2 unit		4 unit	
2x16 real	2 unit	2 unit				2 unit	2 unit					2 unit			
1x16 cmpx															
4x16 real	4 unit			4 unit		4 unit			4 unit					4 unit	
2x16 cmpx															
1x32 real	1 unit	2 unit		4 unit		1unit	2 unit		4 unit		1 unit	2 unit		4 unit	
2x32 real	4 unit	2 unit				2 unit	2 unit					2 unit			
1x32 cmpx															
4x32 real	4 unit			4 unit		4 unit			4 unit		4 unit			4 unit	
2x32 cmpx															
1x40 real	1 unit					1 unit			4 unit		1 unit				
2x40 real	2 unit	2 unit				2 unit	2 unit					2 unit			
1x40 cmpx															
4x40 real	4 unit		4 unit			4 unit			4 unit					4 unit	
2x40 cmpx															

FIG. 15A

	1x16 real	2x16 real	1x16 cmpx	4x16 real	2x16 cmpx	1x32 real	2x32 real	1x32 cmpx	4x32 real	2x32 cmpx	1x40 real	2x40 real	1x40 cmpx	4x40 real	2x40 cmpx
1x16 real	1 unit	2 unit	2 unit	4 unit	4 unit	1 unit	2 unit	2 unit	4 unit	4 unit	1 unit	2 unit	2 unit	4 unit	4 unit
2x16 real	2unit	2 unit				2 unit	2 unit					2 unit			
1x16 cmpx	2unit		2 unit					2 unit			2 unit		2 unit		
4x16 real	4unit			1 unit		4 unit			4 unit					4 unit	
2x16 cmpx	4unit				4 unit					4 unit					4 unit
1x32 real	1unit	2unit				1unit	2 unit	2 unit	4 unit		1 unit	2 unit	2 unit	4 unit	
2x32 real	2unit	2unit				2unit	2 unit					2 unit			
1x32 cmpx	2unit		2unit			2unit		2 unit			2 unit		2 unit		
4x32 real	4unit			4unit		4unit			4 unit		4 unit			4 unit	
2x32 cmpx	4unit				4unit					4 unit					4 unit
1x40 real	1unit		2unit			1unit		2unit	4unit		1 unit	2 unit		4 unit	
2x40 real	2unit	2unit				2unit	2unit				2unit	2 unit			
1x40 cmpx	2unit		2unit			2unit		2unit					2 unit		
4x40 real	4unit			4unit		4unit			4unit		4unit			4 unit	
2x40 cmpx	4unit				4unit					4unit					4 unit

FIG. 15B

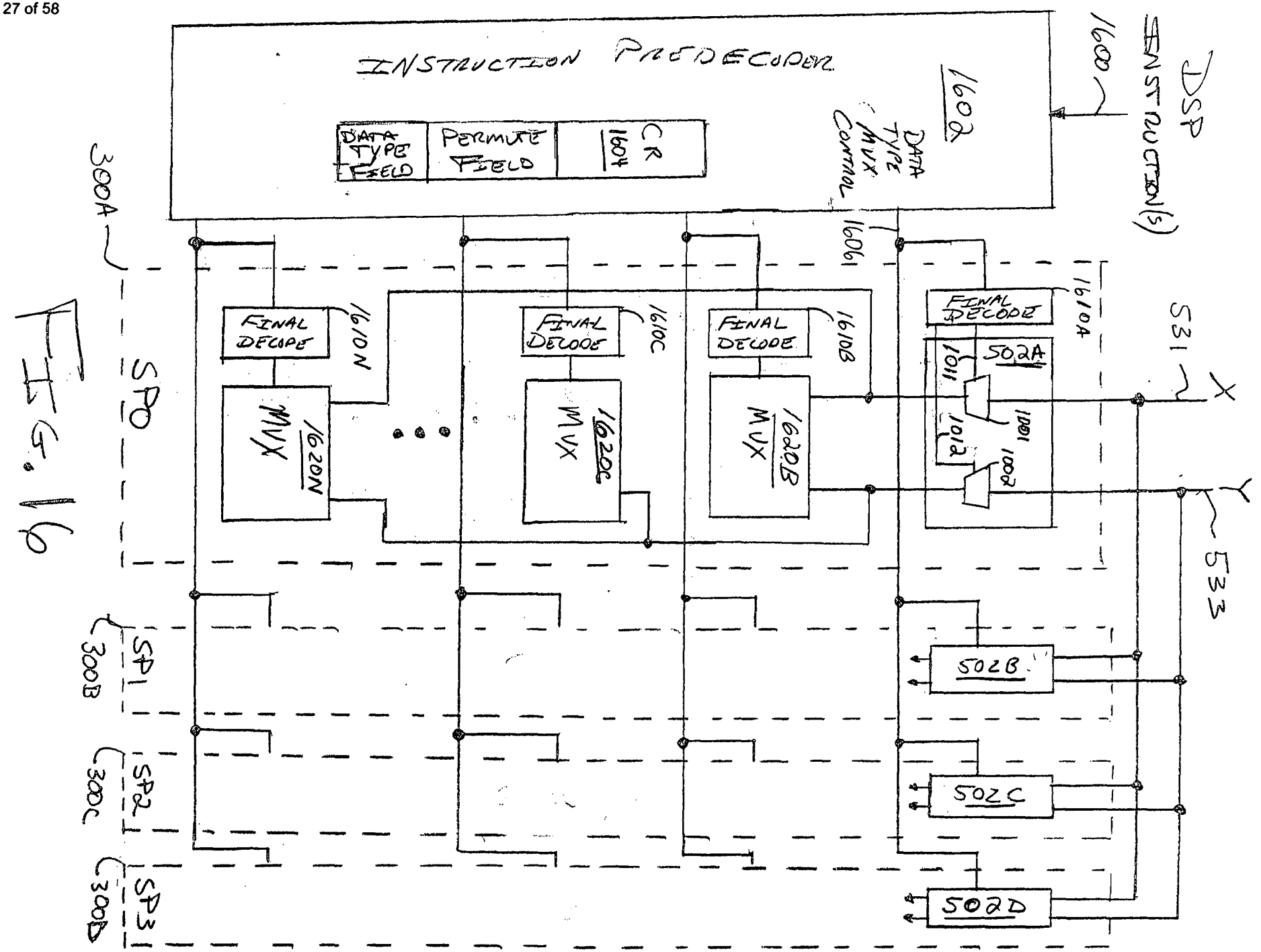

INSTRUCTION PREDECODER
1602
1600
DSP INSTRUCTION(S)
DATA TYPE FIELD
PERMUTE FIELD
CR 1604
DATA TYPE MUX CONTROL
1606
300A
SP0
FINAL DECODE
1610N
1610C
1610B
1610A
MUX 1620N
MUX 1620C
MUX 1620B
502A
1011
1012
1001
1002
531
533
X
Y
SP1 300B
SP2 300C
SP3 300D
502B
502C
502D

FIG. 16

Data Type: N x S(R/C)

FIG. 17

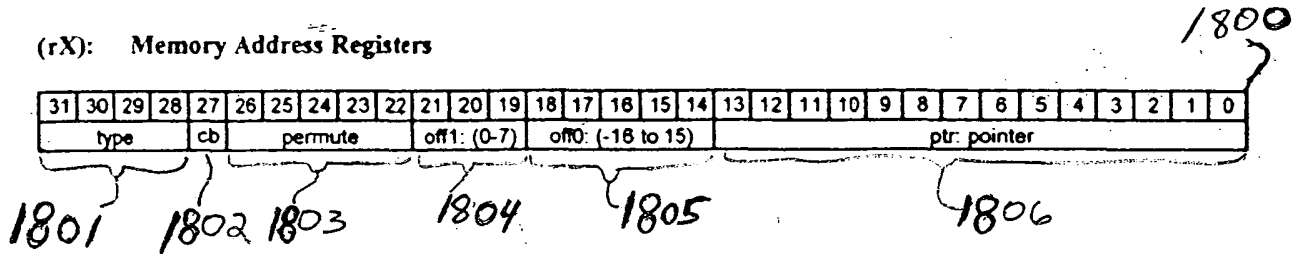


FIG. 18

DATA TYPE 1801

0000: 1x16 real
0001: 2x16 real
0010: 1x16 complex
0011: 4x16 real
0100: 1x32 real
0101: 2x32 real
0110: 1x32 complex
0111: 2X16 complex
1000: 4x32 real
1001: 2x32 complex
1010: 1x40 real
1011: 2x40 real
1100: 1x40 complex
1101: 4x40 real (only for local add unit operations)
1110: 2x40 complex (only for local add unit operations)
1111: Reserved

FIG. 19

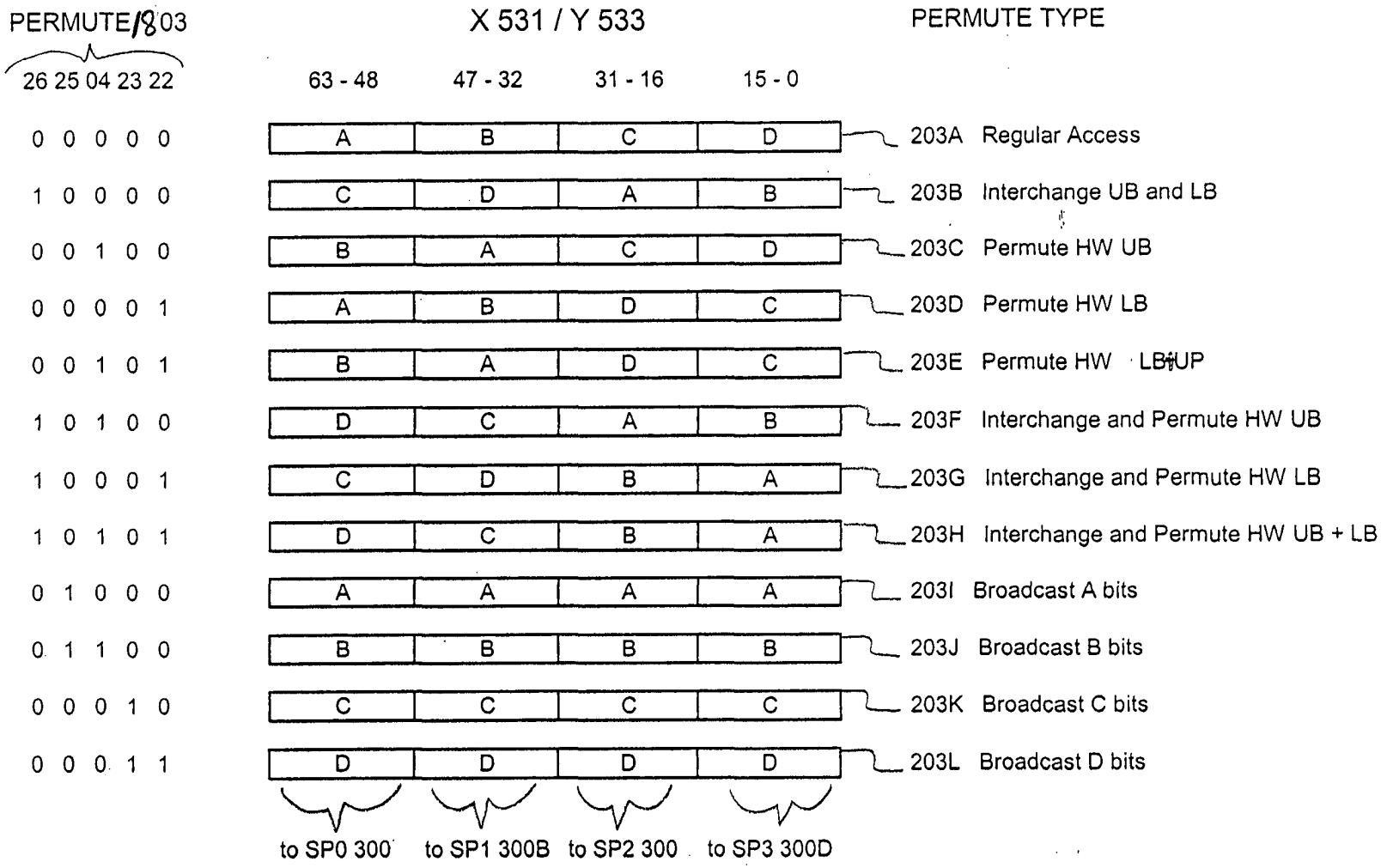

PERMUTE 803
26 25 04 23 22
X 531 / Y 533
63 - 48
47 - 32
31 - 16
15 - 0
PERMUTE TYPE
0 0 0 0 0
A B C D
203A Regular Access
1 0 0 0 0
C D A B
203B Interchange UB and LB
0 0 1 0 0
B A C D
203C Permute HW UB
0 0 0 0 1
A B D C
203D Permute HW LB
0 0 1 0 1
B A D C
203E Permute HW LB&UP
1 0 1 0 0
D C A B
203F Interchange and Permute HW UB
1 0 0 0 1
C D B A
203G Interchange and Permute HW LB
1 0 1 0 1
D C B A
203H Interchange and Permute HW UB + LB
0 1 0 0 0
A A A A
203I Broadcast A bits
0 1 1 0 0
B B B B
203J Broadcast B bits
0 0 0 1 0
C C C C
203K Broadcast C bits
0 0 0 1 1
D D D D
203L Broadcast D bits
to SP0 300
to SP1 300B
to SP2 300
to SP3 300D

FIG. 20

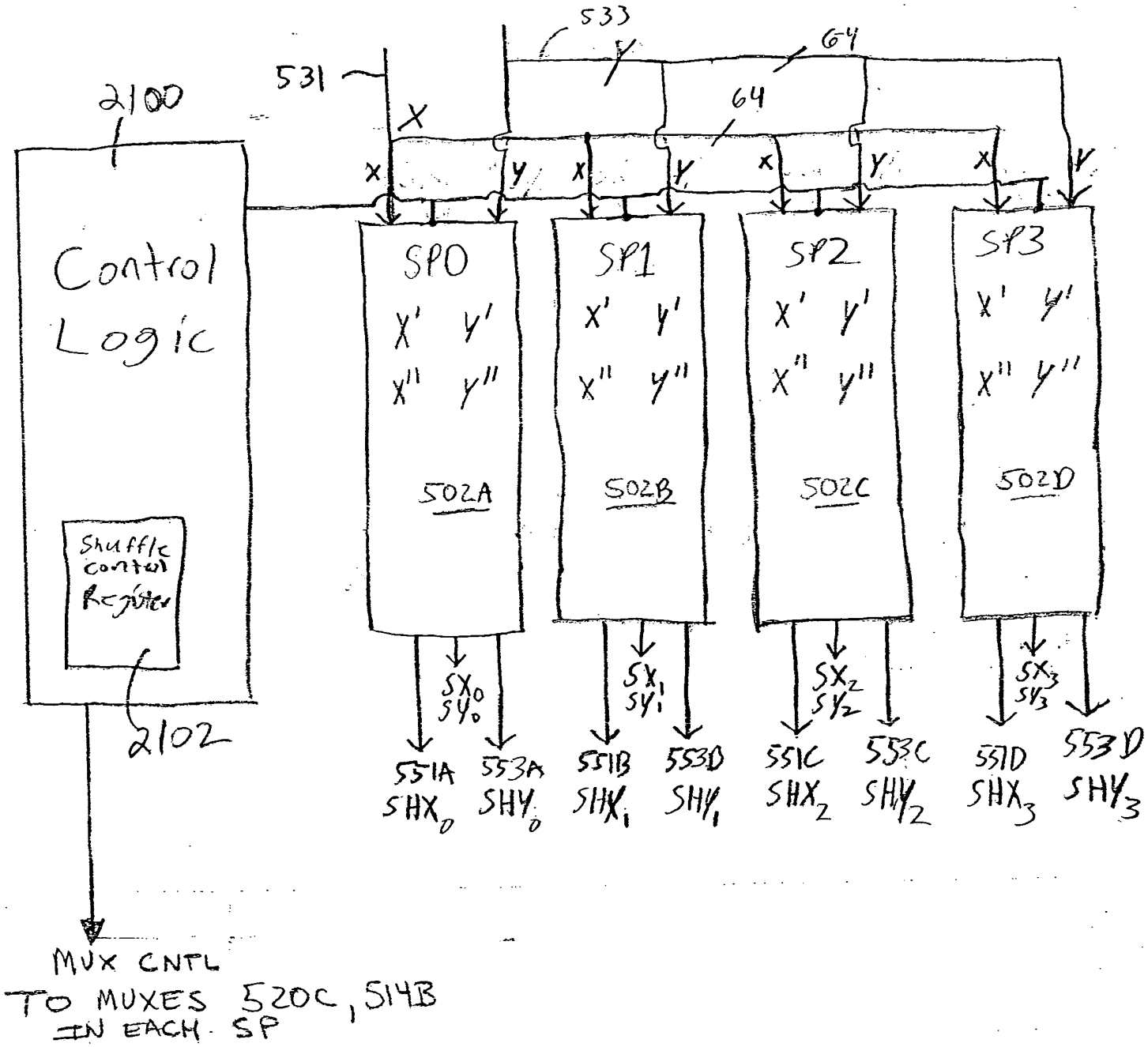

2100
Control Logic
Shuffle control Register
2102
MUX CNTL
TO MUXES 520C, 514B IN EACH SP
531
533
64
64
X
x
y
SP0
SP1
SP2
SP3
x' y'
x'' y''
502A
502B
502C
502D
SX0 SY0
SX1 SY1
SX2 SY2
SX3 SY3
551A
553A
551B
553B
551C
553C
551D
553D
SHX0
SHY0
SHX1
SHY1
SHX2
SHY2
SHX3
SHY3

FIG. 21

$X' = [SX_{10}, SX_{11}, SX_{12}, SX_{13}]$ e.g. $[X_0, X_1, X_2, X_3]$

$X'' = [SX_{20}, SX_{21}, SX_{22}, SX_{23}]$ e.g. $[X_4, X_5, X_6, X_7]$

Where SX_{ab} : S = Source; a = delay; b = SP unit number (e.g. SP3, SP2, SP1, SP0; or termed U3, U2, U1, U0)

$Y' = [SY_{10}, SY_{11}, SY_{12}, SY_{13}]$

$Y'' = [SY_{20}, SY_{21}, SY_{22}, SY_{23}]$

Where SY_{ab} : S = Source; a = delay; b = SP unit number (e.g. SP3, SP2, SP1, SP0; or termed U_3, U_2, U_1, U_0)

FIG. 22A

---

shuffle **Shuffle Control Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
u3		u2		u1		u0		u3		u2		u1		u0		u3		u2		u1		u0		u3		u2		u1		u0	
SY2S								SY1S								SX2S								SX1S							

Units are connected to their nearest neighbors for shuffling the sources using the following bit diagram:

00 Unit N+1, SX1 = X' (right)
01 Unit N+1, SX2 = X'' (right)
10 Unit N-1, SX1 = X' (left)
11 Unit N-1, SX2 = X'' (left)

For example to shift the sources to the left by one:

3	2	1	0	From
2	1	0	3	Into

The bits should be 10101010 ($AA)

FIG. 22C

FIR Filter $\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_N \end{bmatrix} * \begin{bmatrix} Y_0 \\ \vdots \\ Y_N \end{bmatrix} = X_0Y_0 + X_1Y_1 \ldots X_NY_N$

Primary Stage

Primary Stage Computations

Cycle #	SP0	SP1	SP2	SP3
1	X_0Y_0 +	X_1Y_1 +	X_2Y_2 +	X_3Y_3
2	X_4Y_4 +	X_5Y_5 +	X_6Y_6 +	X_7Y_7
3	X_8Y_8 +	X_9Y_9 +	$X_{10}Y_{10}$ +	$X_{11}Y_{11}$
⋮		⋮		
N	$X_{N-3}Y_{N-3}$ +	$X_{N-2}Y_{N-2}$ +	$Y_{N-1}Y_{N-1}$ +	X_NY_N

Shadow Stage

Shadow Stage Computations

Cycle #	SP0	SP1	SP2	SP3
1	No operation			
2	No operation			
3	X_1Y_0 +	X_2Y_1 +	X_3Y_2 +	X_4Y_3
4	X_5Y_4 +	X_6Y_5 +	X_7Y_6 +	X_8Y_7
⋮		⋮		
N+2	$X_{N-2}Y_{N-3}$ +	$X_{N-1}Y_{N-2}$ +	X_NY_{N-1} +	$X_{N+1}Y_N$

$\begin{bmatrix} X_1 \\ \vdots \\ X_{N+1} \end{bmatrix} * \begin{bmatrix} Y_0 \\ \vdots \\ Y_N \end{bmatrix}$ (Shuffle X' Left by one)

Subsequent Cycles

Primary Stage

Cycle #		
N+1 ⋮ 2N	$\begin{bmatrix} X_2 \\ \vdots \\ X_{N+2} \end{bmatrix}$	$* \begin{bmatrix} Y_0 \\ \vdots \\ Y_N \end{bmatrix}$
N+4 ⋮ 3N	$\begin{bmatrix} X_4 \\ \vdots \\ X_{N+4} \end{bmatrix}$	$* \begin{bmatrix} Y_0 \\ \vdots \\ Y_N \end{bmatrix}$

Shadow Stage

Cycle #		
N+3 ⋮ N+5	$\begin{bmatrix} X_3 \\ \vdots \\ X_{N+3} \end{bmatrix}$	$* \begin{bmatrix} Y_0 \\ \vdots \\ Y_N \end{bmatrix}$
N+5 ⋮ N+7	$\begin{bmatrix} X_5 \\ \vdots \\ X_{N+5} \end{bmatrix}$	$* \begin{bmatrix} Y_0 \\ \vdots \\ Y_N \end{bmatrix}$

⋮

FIG. 22B

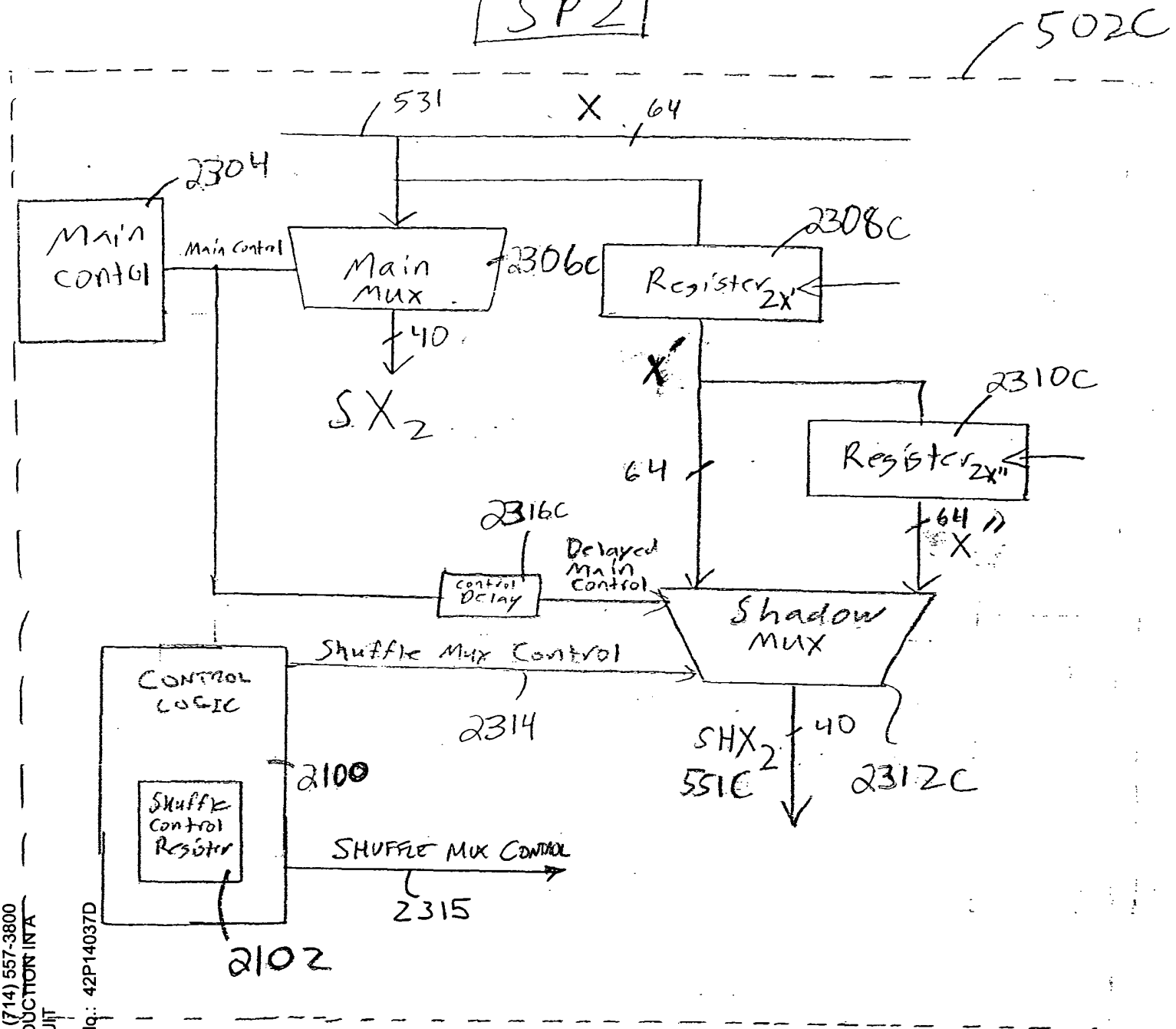


FIG. 23A

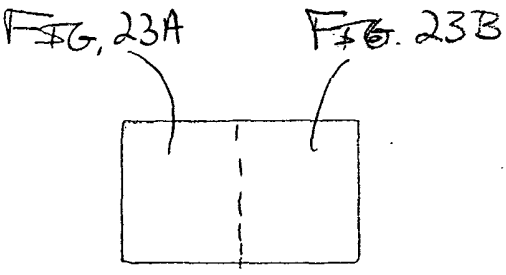


FIG 23

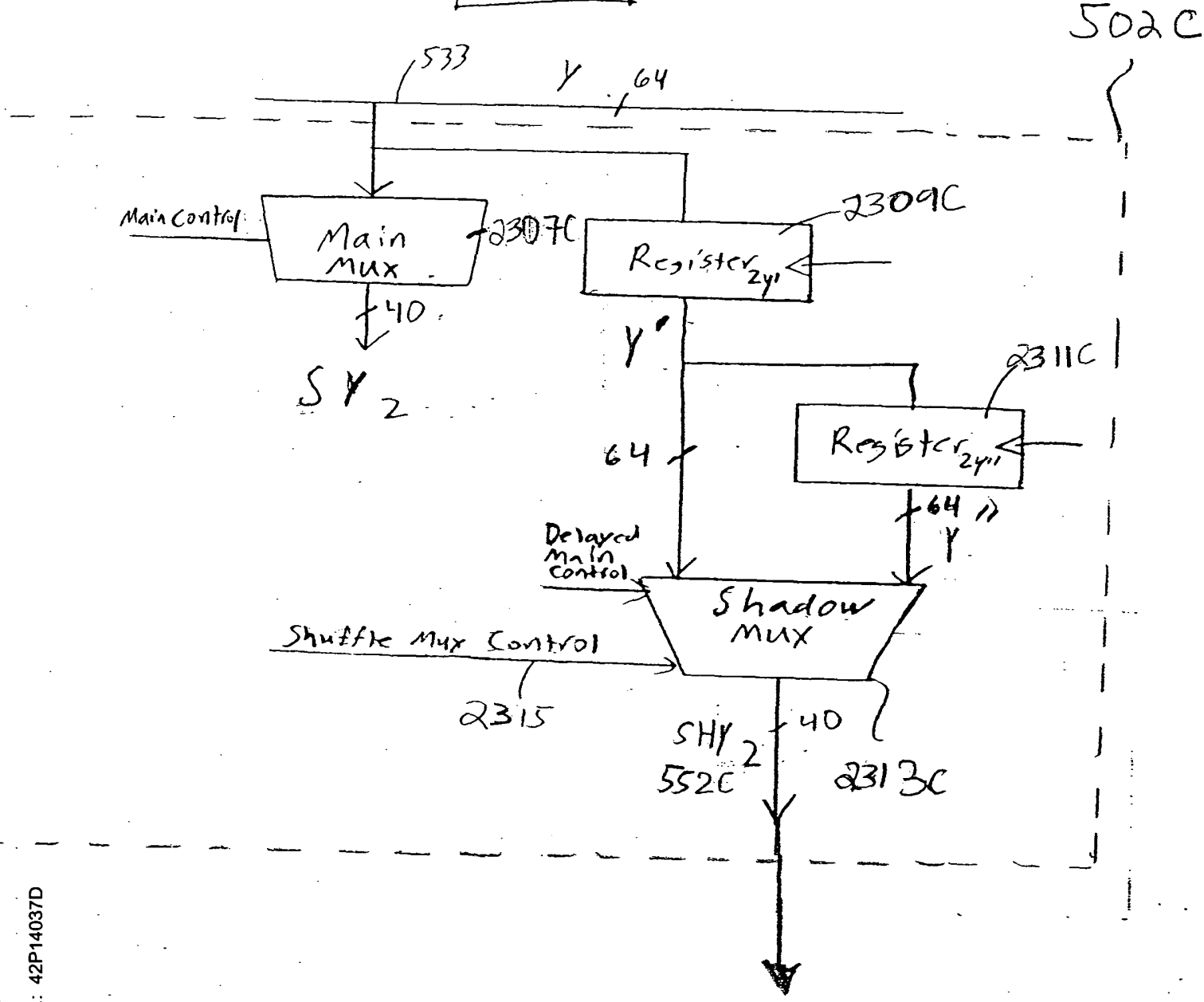
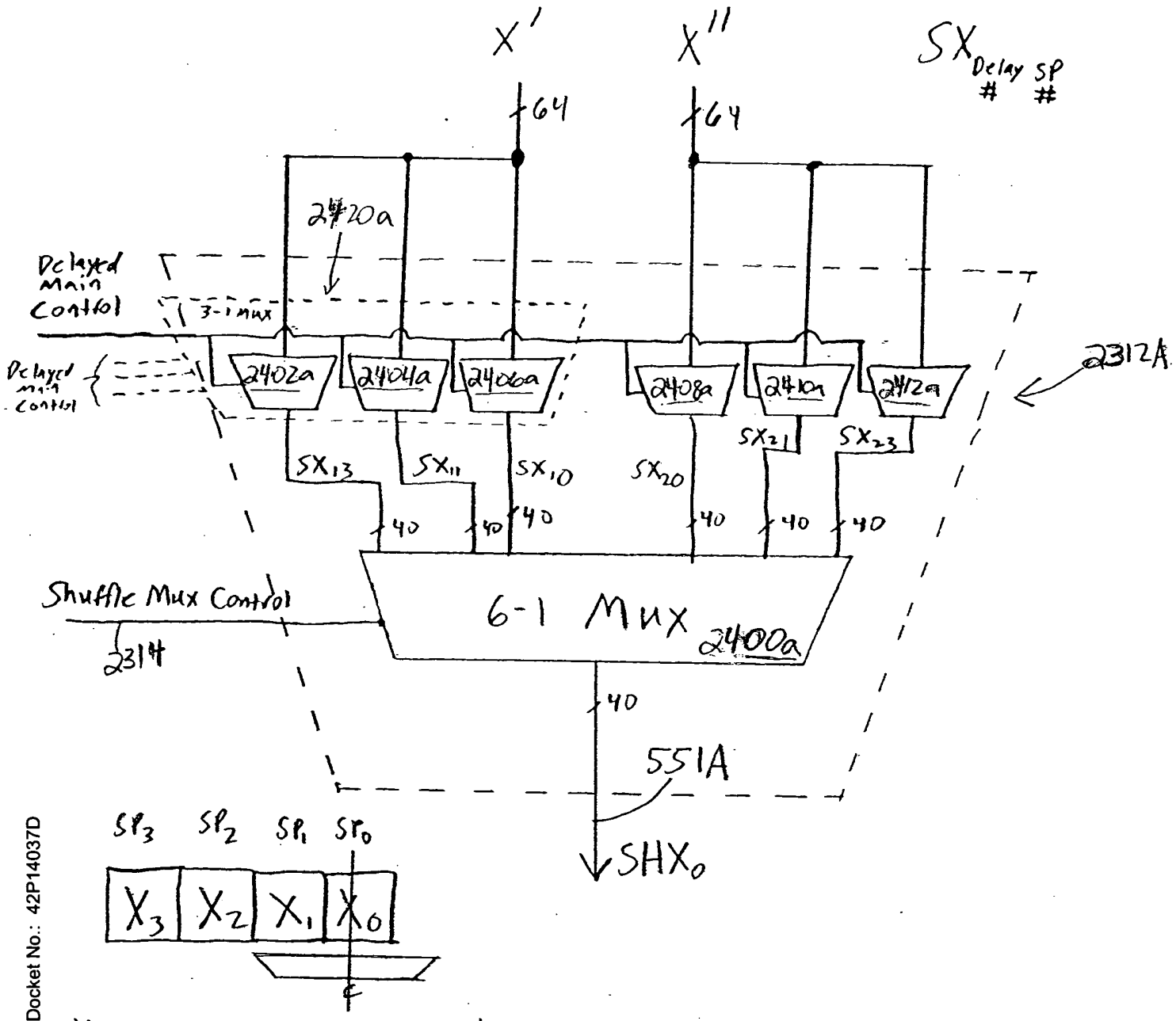


FIG. 23B

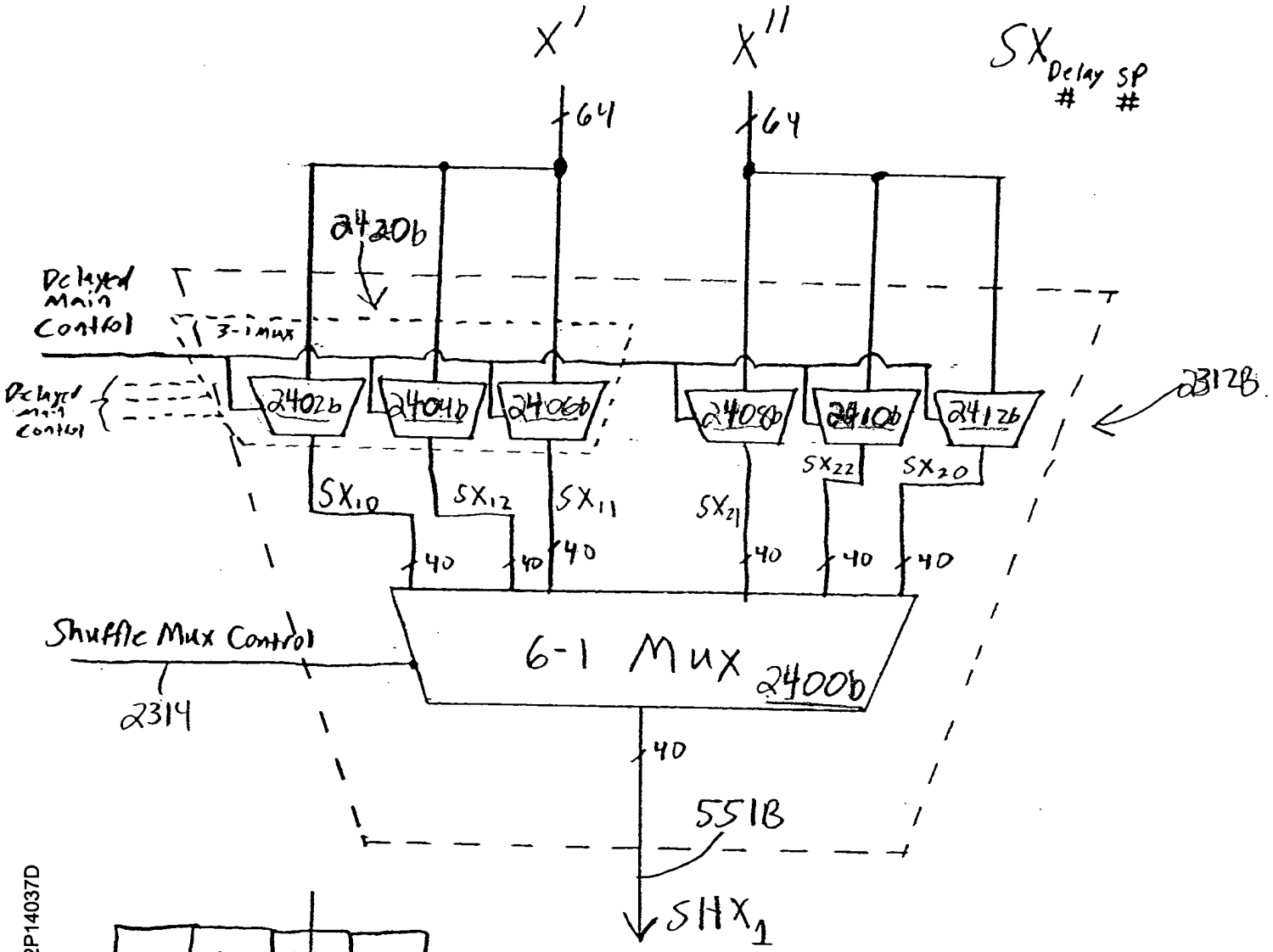
SPO
Shadow
Mux

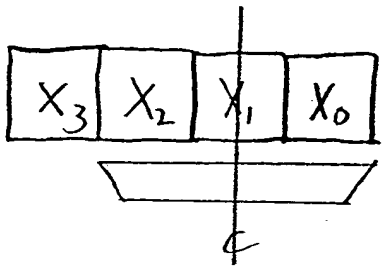


$X_0 = SX_{10}, SX_{20}$
$X_1 = SX_{11}, SX_{21}$
$X_3 = SX_{13}, SX_{23}$

FIG. 24A

SP1
Shadow
Mux





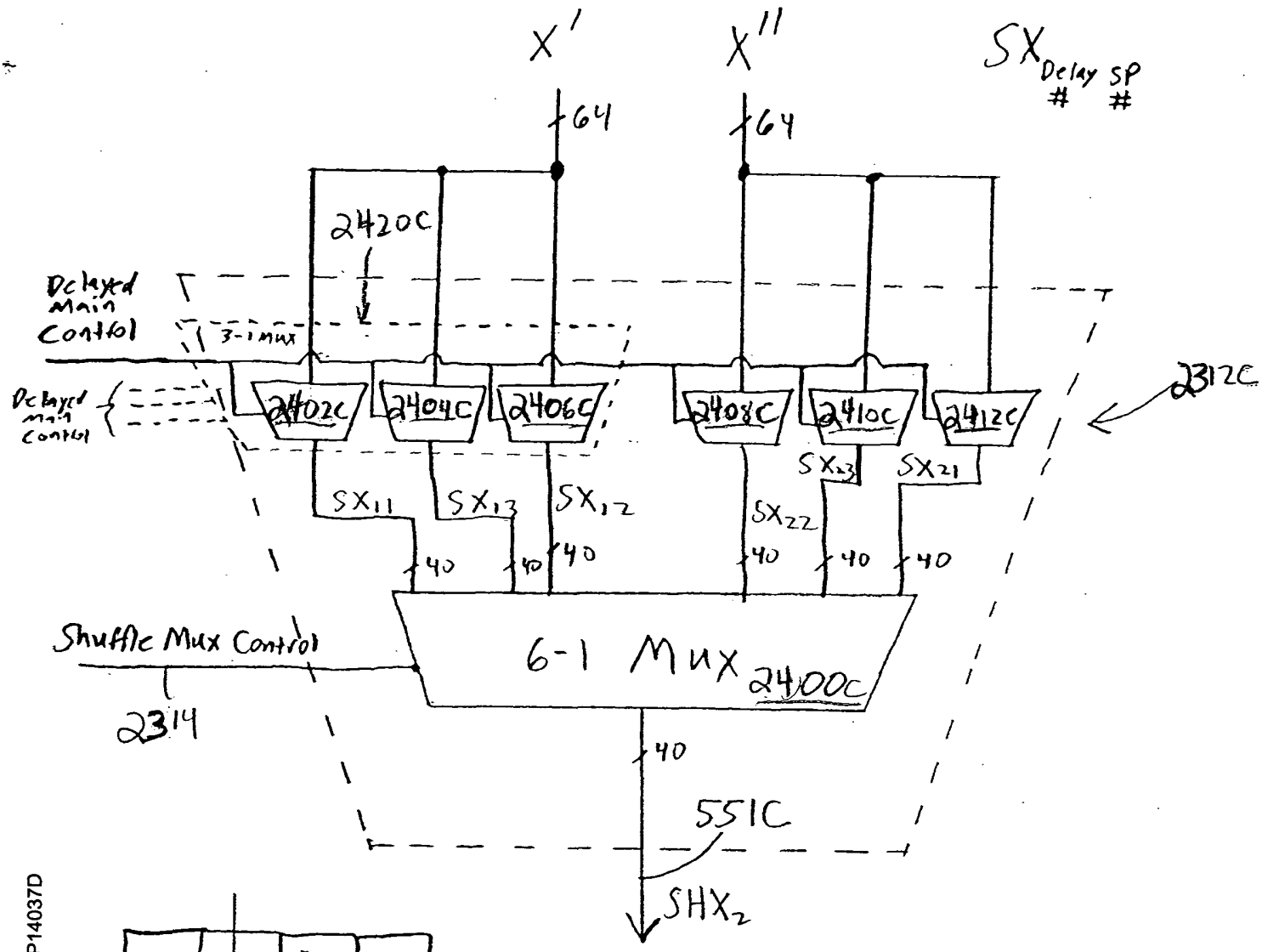
$X_1 = SX_{11}, SX_{21}$

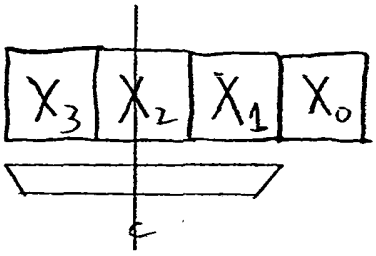
$X_2 = SX_{12}, SX_{22}$

$X_0 = SX_{10}, SX_{20}$

FIG. 24B

SP2
Shadow
Mux

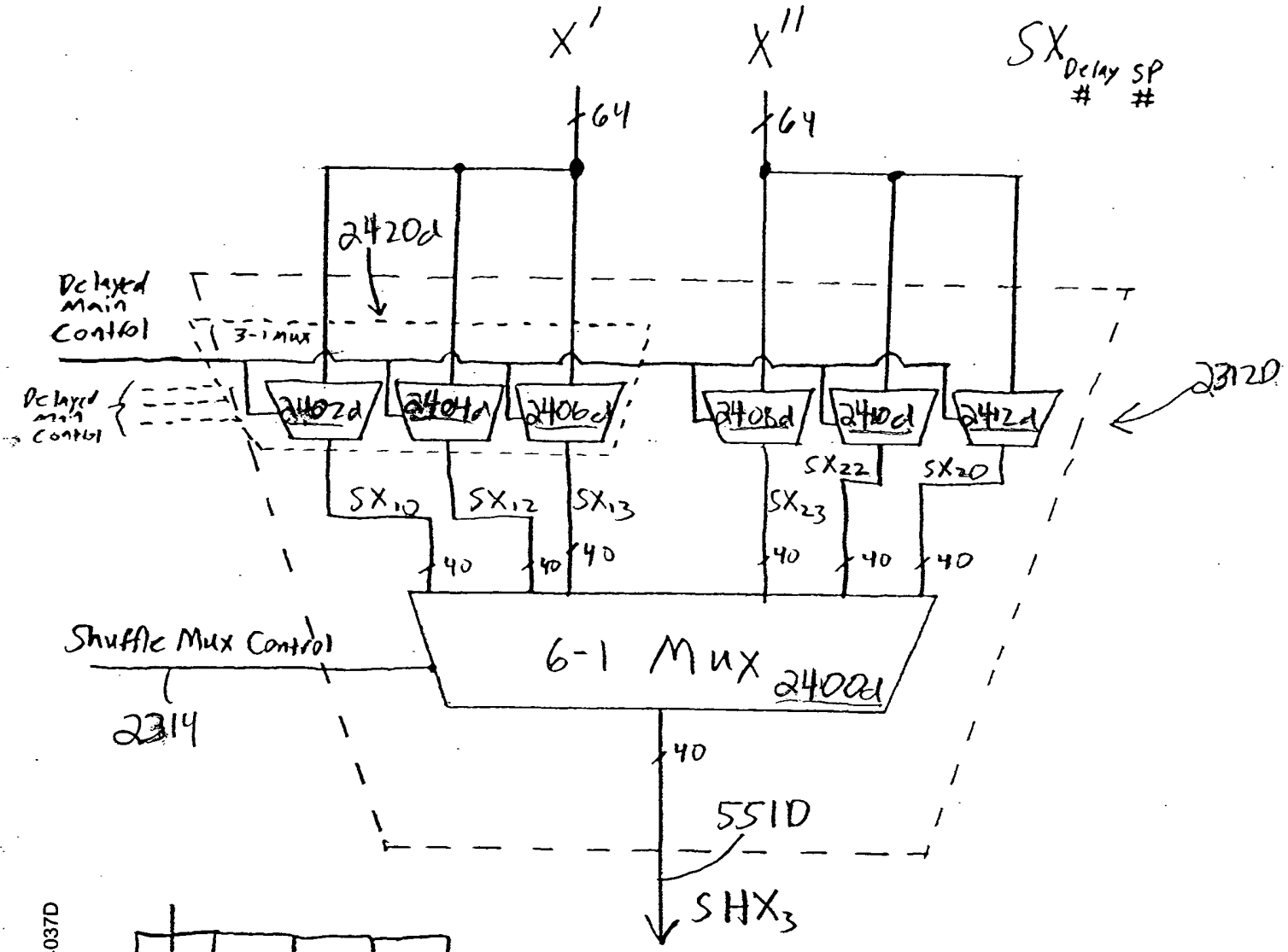




$X_2 = SX_{12}, SX_{22}$
$X_3 = SX_{13}, SX_{23}$
$X_1 = SX_{11}, SX_{21}$

FIG. 24C

SP3
Shadow
Mux



$X_3 = SX_{13}, SX_{23}$

$X_0 = SX_{10}, SX_{20}$

$X_2 = SX_{12}, SX_{22}$

FIG. 24D

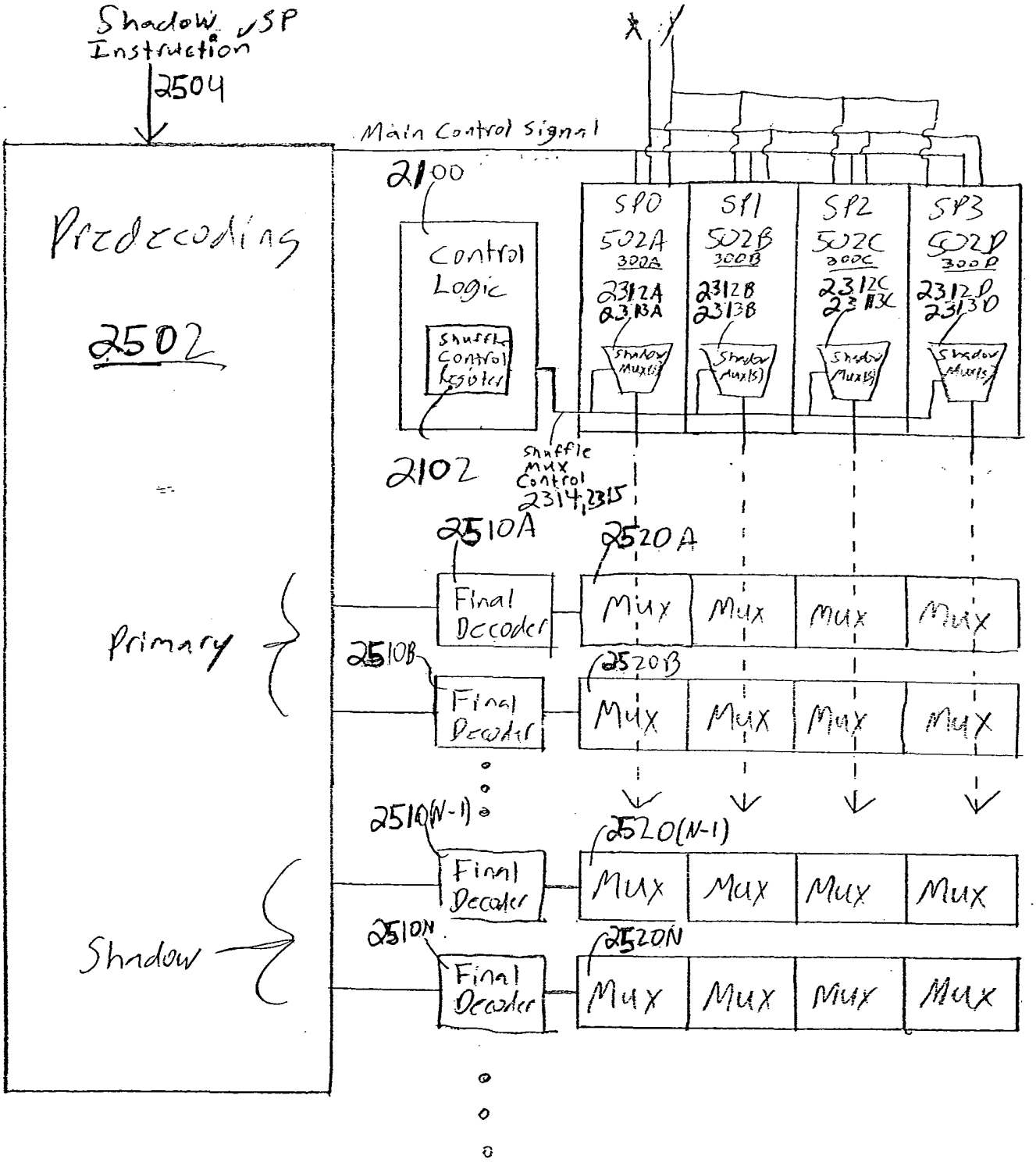


FIG. 25

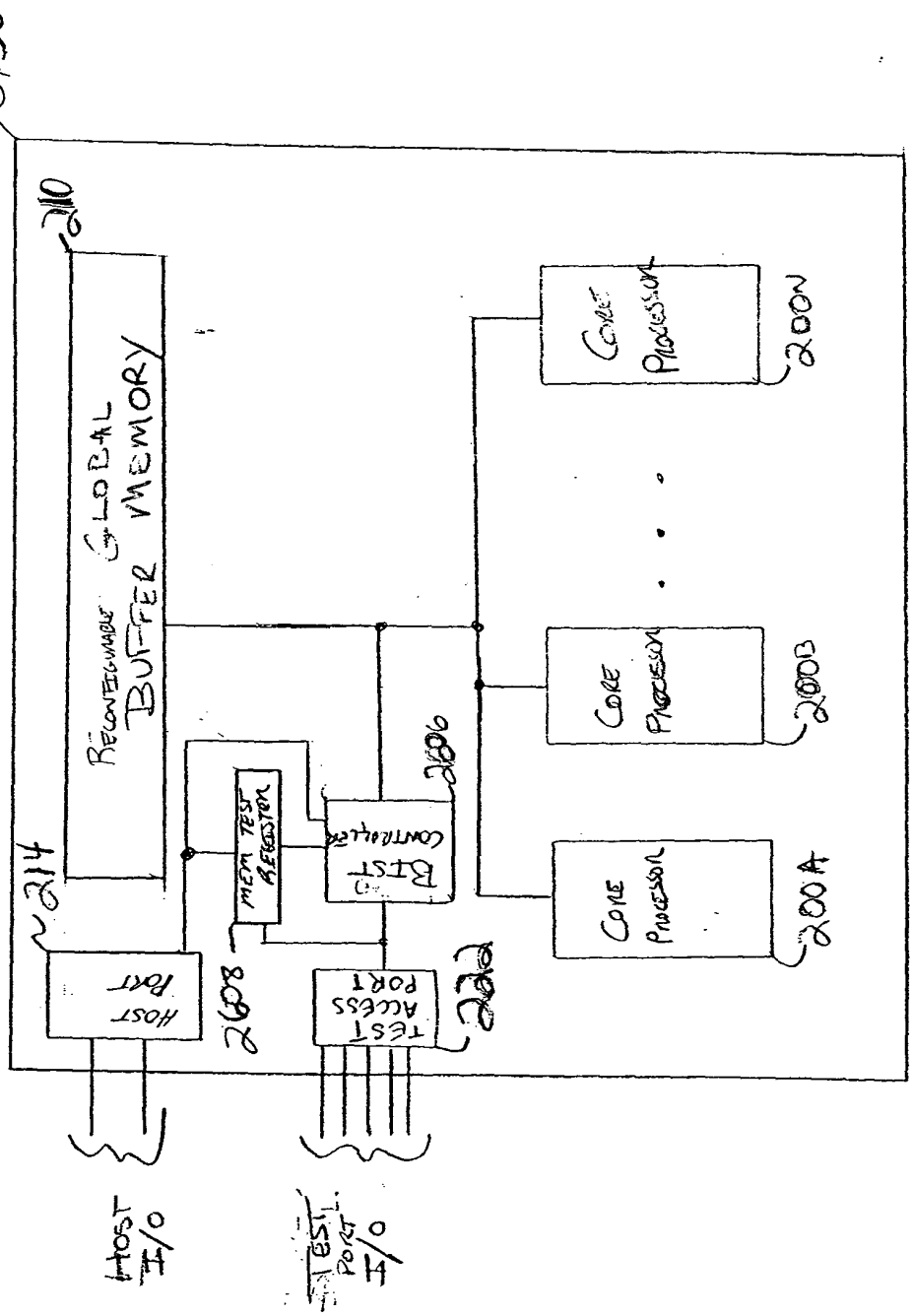

150
110
RECONFIGURABLE GLOBAL BUFFER MEMORY
214
HOST PORT
208
MEM TEST REGISTER
206
BIST CONTROLLER
202
TEST ACCESS PORT
200A
CORE PROCESSOR
200B
CORE PROCESSOR
200N
CORE PROCESSOR
HOST I/O
TEST PORT I/O

FIG. 2G

210

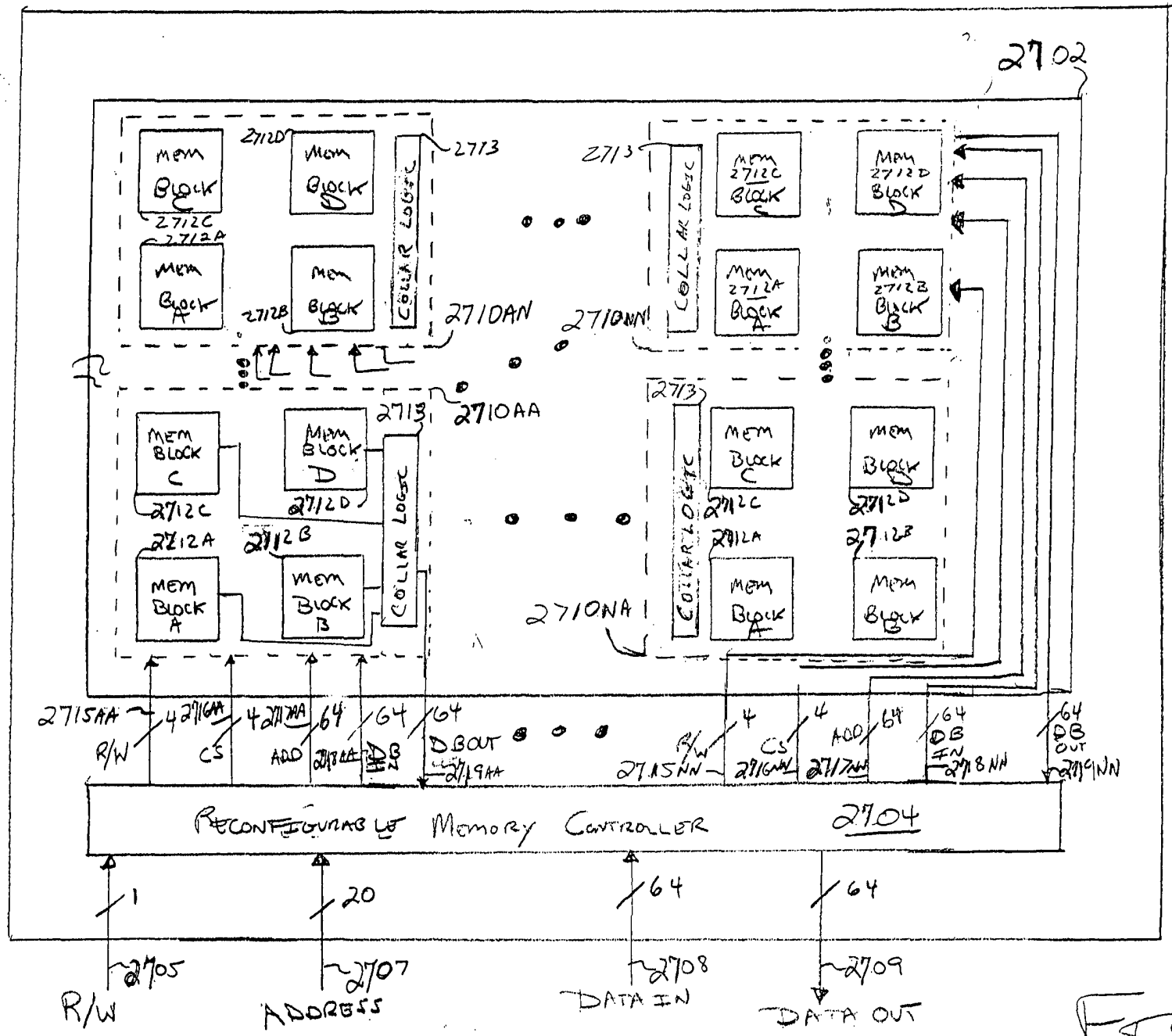

2702
MEM BLOCK C
MEM BLOCK D
MEM BLOCK A
MEM BLOCK B
2712C
2712D
2712A
2712B
2713
COLLAR LOGIC
2710AN
2710NN
2710AA
2710NA
2715AA
2716AA
2717AA
2718AA
2719AA
2715NN
2716NN
2717NN
2718NN
2719NN
R/W
CS
ADD
DB IN
DB OUT
RECONFIGURABLE MEMORY CONTROLLER
2704
2705
2707
2708
2709
ADDRESS
DATA IN
DATA OUT

FIG. 27

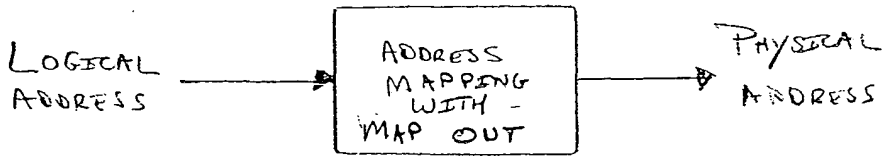

LOGICAL ADDRESS
ADDRESS MAPPING WITH - MAP OUT
PHYSICAL ADDRESS

FIG. 28

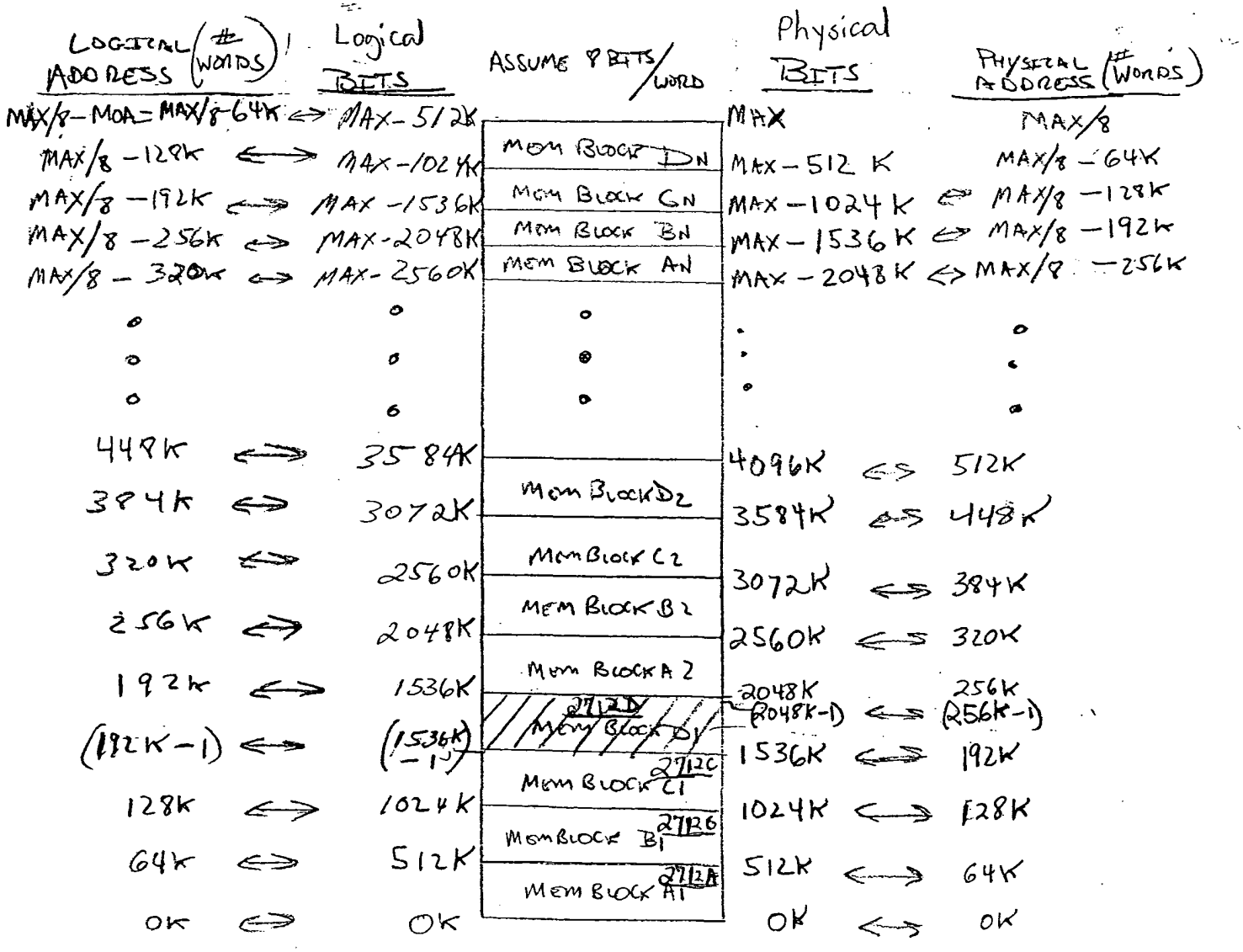

LOGICAL ADDRESS (# WORDS)
Logical BITS
ASSUME 8 BITS/WORD
Physical BITS
PHYSICAL ADDRESS (# WORDS)
MAX/8 - MOA = MAX/8 - 64K ⟷ MAX - 512K
MAX/8 - 128K ⟷ MAX - 1024K
MAX/8 - 192K ⟷ MAX - 1536K
MAX/8 - 256K ⟷ MAX - 2048K
MAX/8 - 320K ⟷ MAX - 2560K
448K ⟷ 3584K
384K ⟷ 3072K
320K ⟷ 2560K
256K ⟷ 2048K
192K ⟷ 1536K
(192K - 1) ⟷ (1536K - 1)
128K ⟷ 1024K
64K ⟷ 512K
0K ⟷ 0K
MAX
MAX - 512 K
MAX - 1024 K ⟷ MAX/8 - 128K
MAX - 1536 K ⟷ MAX/8 - 192K
MAX - 2048 K ⟷ MAX/8 - 256K
4096K ⟷ 512K
3584K ⟷ 448K
3072K ⟷ 384K
2560K ⟷ 320K
2048K (2048K-1) ⟷ 256K (256K-1)
1536K ⟷ 192K
1024K ⟷ 128K
512K ⟷ 64K
0K ⟷ 0K
MEM BLOCK DN
MEM BLOCK GN
MEM BLOCK BN
MEM BLOCK AN
MEM BLOCK D2
MEM BLOCK C2
MEM BLOCK B2
MEM BLOCK A2
MEM BLOCK D1
MEM BLOCK C1
MEM BLOCK B1
MEM BLOCK A1
27126
27126
27126
27126

FIG. 29

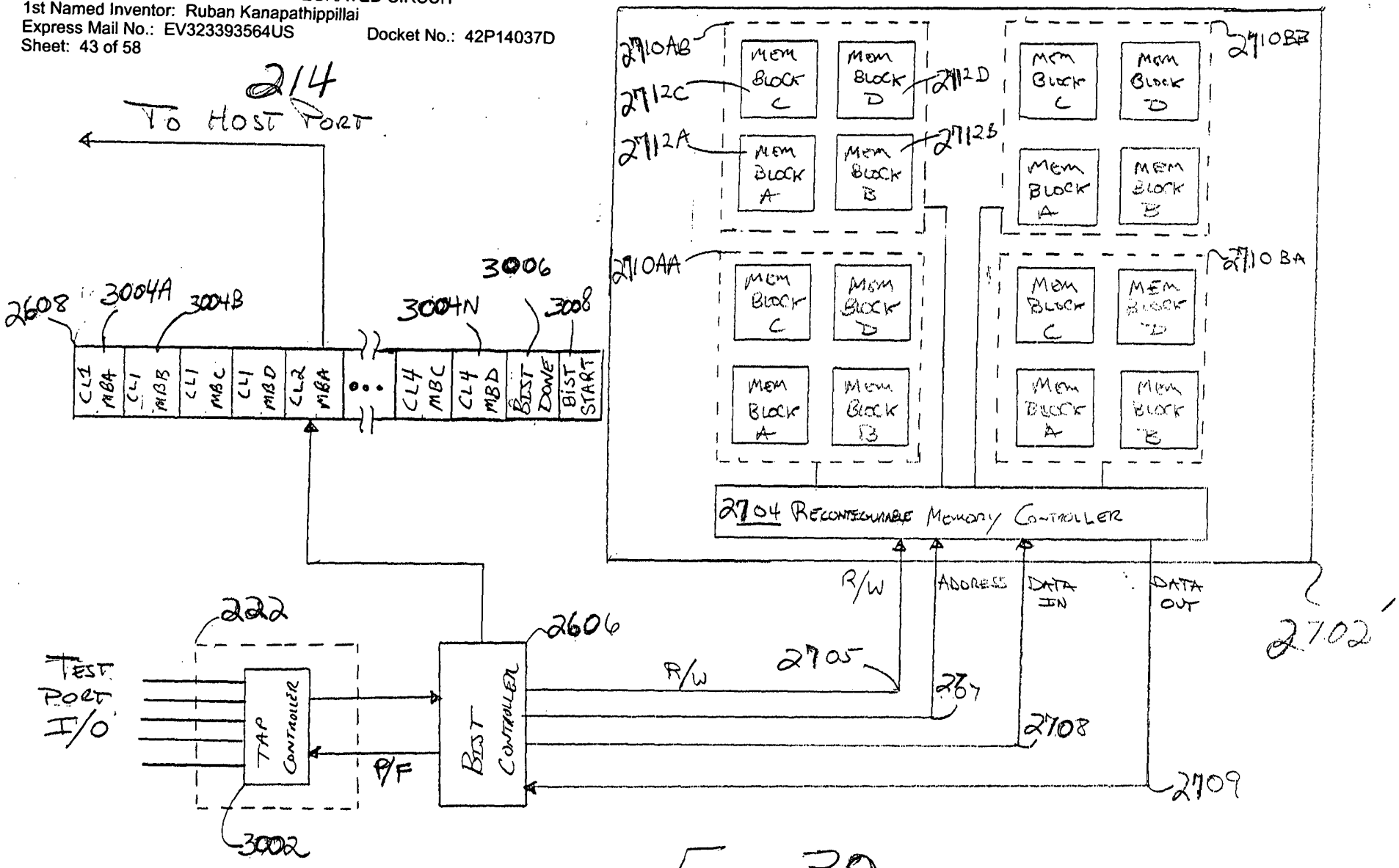

214
To Host Port
2608
3004A
3004B
3006
3004N
3008
CL1 MBA
CL1 MBB
CL1 MBC
CL1 MBD
CL2 MBA
CL4 MBC
CL4 MBD
BIST DONE
BIST START
2710AB
2712C
2712D
2712A
2712B
2710BB
2710AA
2710BA
MEM BLOCK A
MEM BLOCK B
MEM BLOCK C
MEM BLOCK D
MEM BLOCK E
2704 Reconfigurable Memory Controller
R/W
ADDRESS
DATA IN
DATA OUT
2702'
222
2606
TEST PORT I/O
TAP CONTROLLER
BIST CONTROLLER
P/F
2705
2707
2708
2709
3002

FIG. 30

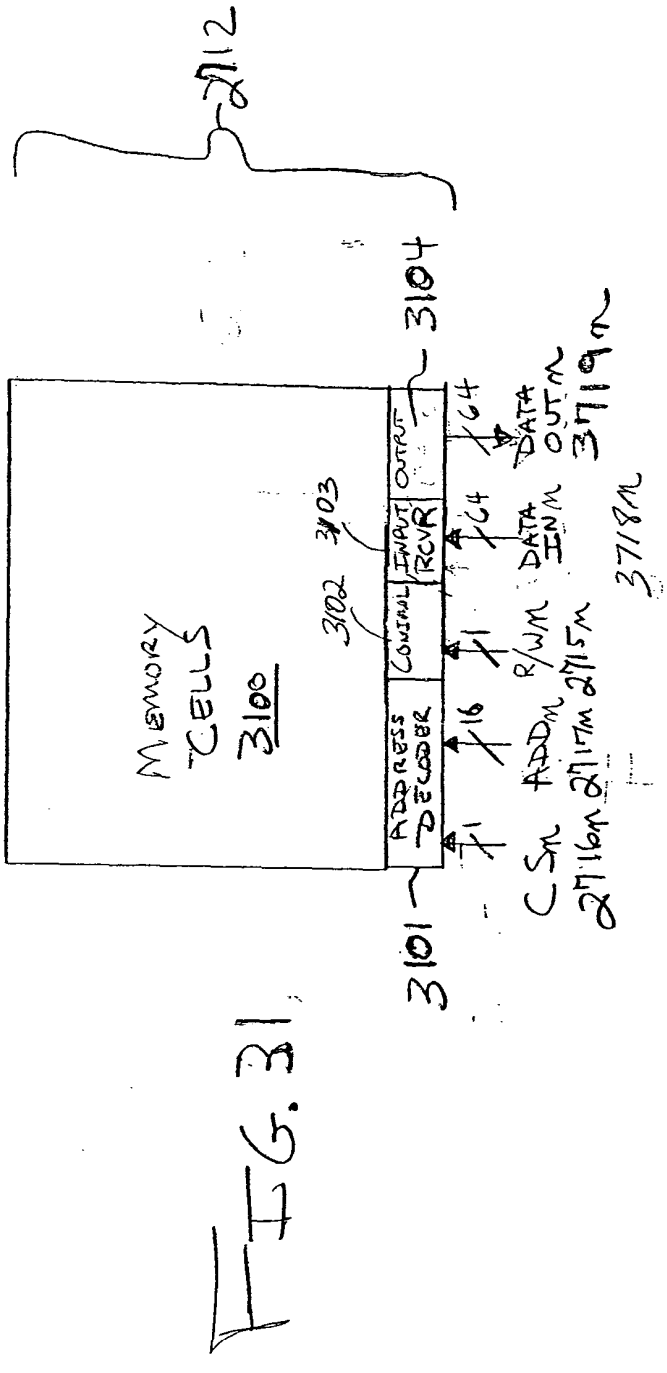

MEMORY CELLS 3100
ADDRESS DECODER
CONTROL
INPUT RCVR
OUTPUT
3101
3102
3103
3104
3112
64
64
16
1
1
DATA OUT M
DATA IN M
R/W M
ADD M
CS M
3118M
3119M

FIG. 31

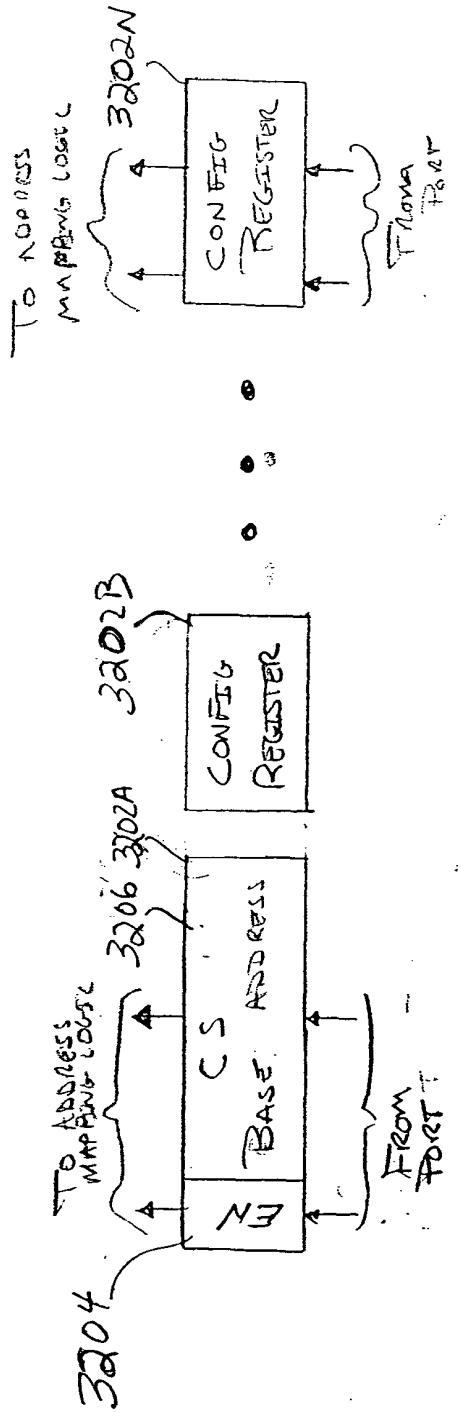

TO ADDRESS MAPPING LOGIC
3202N
CONFIG REGISTER
FROM PORT
3202B
CONFIG REGISTER
3206
3202A
CS
BASE ADDRESS
EN
TO ADDRESS MAPPING LOGIC
FROM PORT
3204

FIG. 32

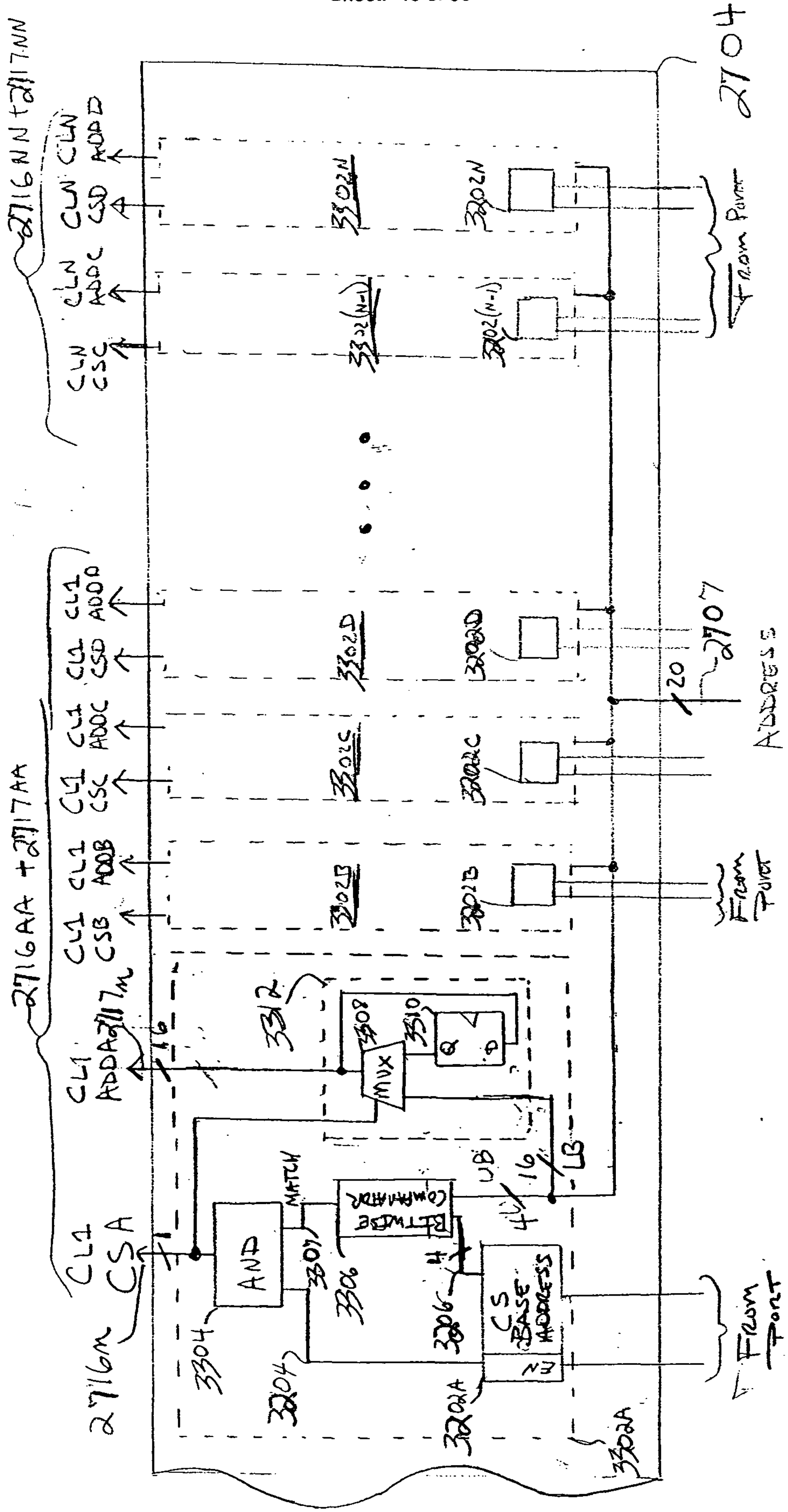

AND
MUX
CS BASE ADDRESS
EN
BITWISE COMPARATOR
MATCH
From Port
ADDRESS
2704
2707
3304
3306
3308
3310
3312

FIG. 33A

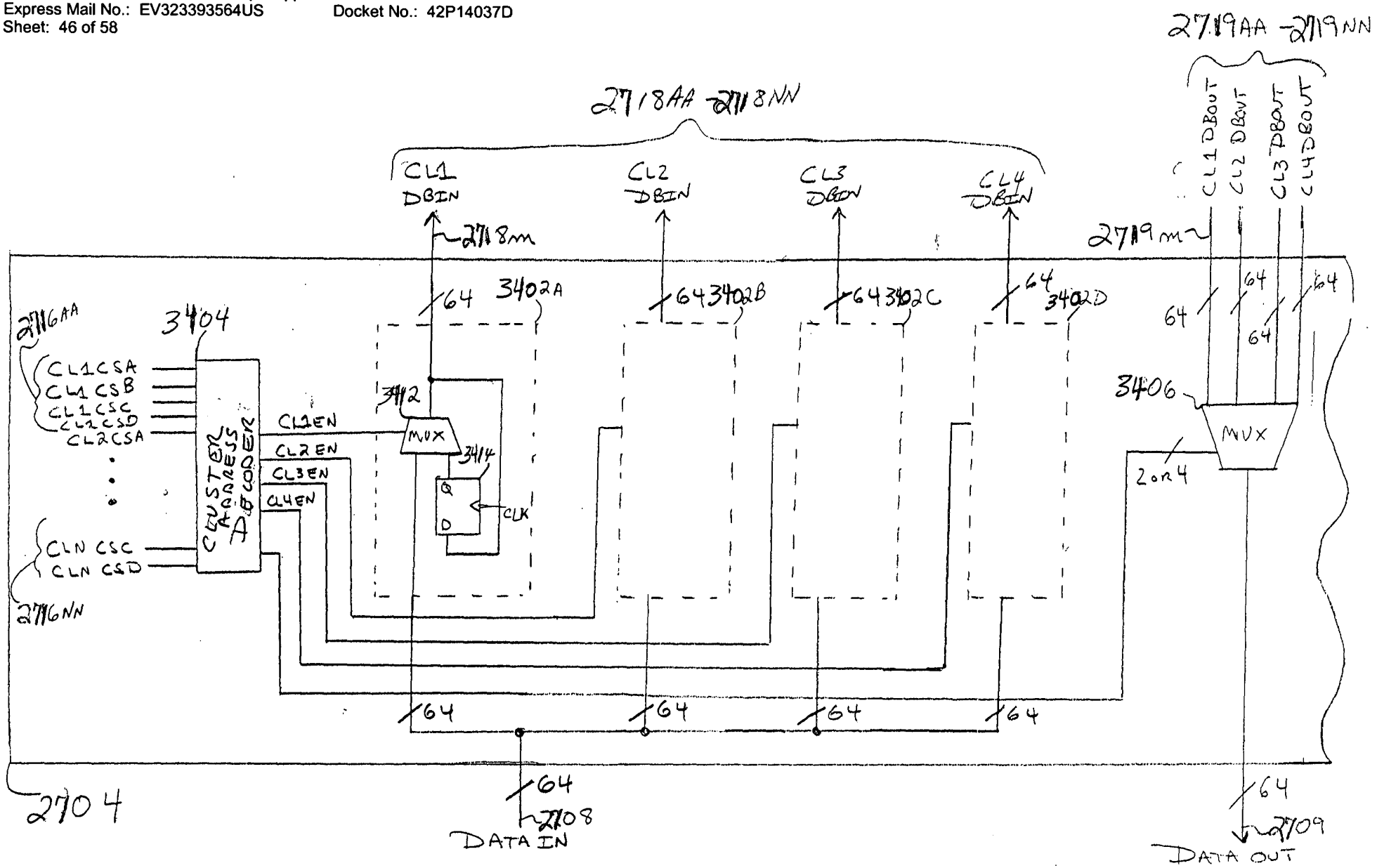

2719AA - 2719NN
2718AA - 2718NN
CL1 DBIN
CL2 DBIN
CL3 DBIN
CL4 DBIN
2718m
2719m
CL1 DBOUT
CL2 DBOUT
CL3 DBOUT
CL4 DBOUT
64
3402A
3402B
3402C
3402D
2716AA
3404
CL1CSA
CL1CSB
CL1CSC
CL2CSD
CL2CSA
CLNCSC
CLNCSD
2716NN
CLUSTER ADDRESS DECODER
CL1EN
CL2EN
CL3EN
CL4EN
3412
MUX
3414
Q
D
CLK
3406
2or4
2704
2708
DATA IN
2709
DATA OUT

FIG. 33B

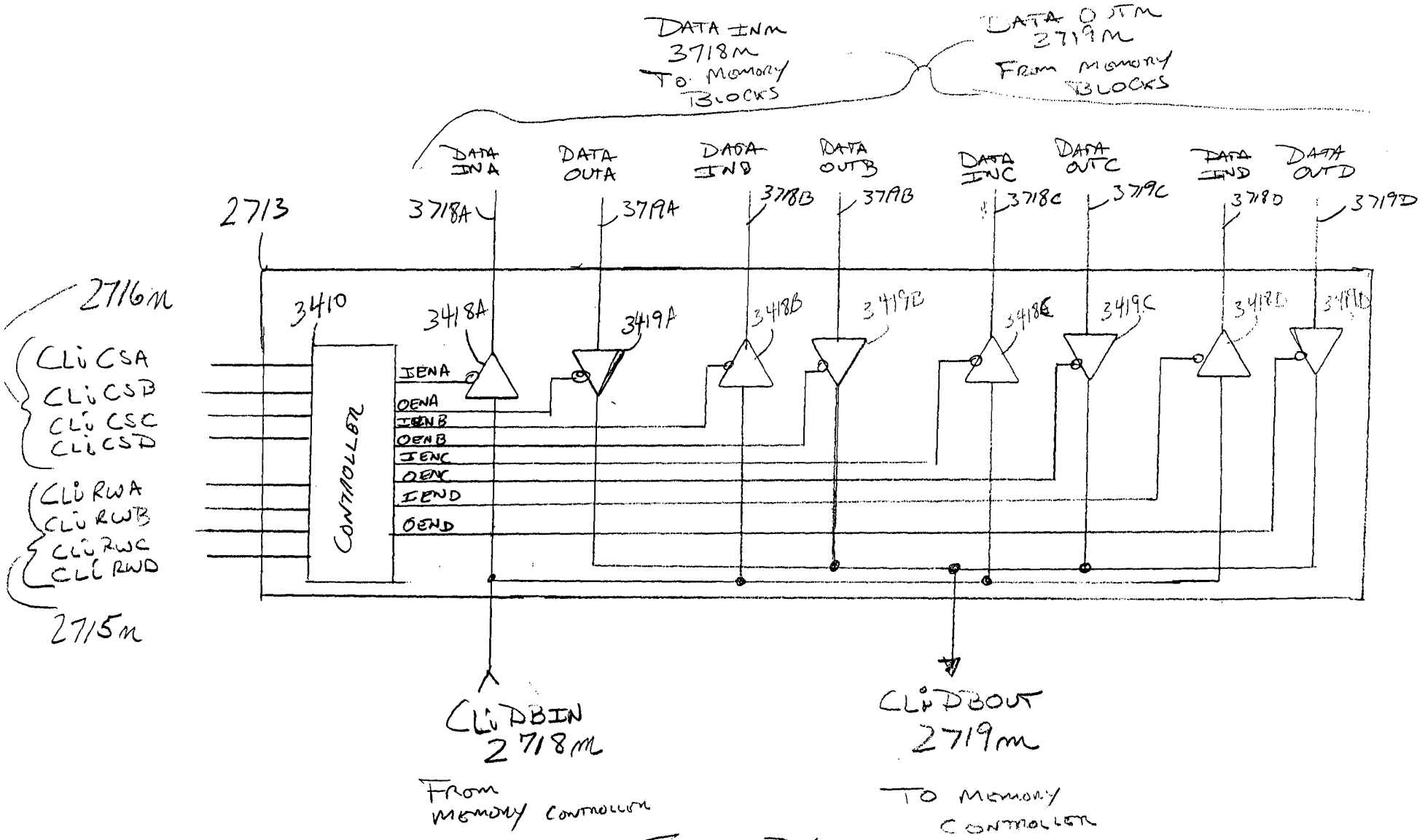

DATA INM 3718M TO MEMORY BLOCKS
DATA OUTM 3719M FROM MEMORY BLOCKS
DATA INA
DATA OUTA
DATA INB
DATA OUTB
DATA INC
DATA OUTC
DATA IND
DATA OUTD
2713
3718A
3719A
3718B
3719B
3718C
3719C
3718D
3719D
2716M
3410
3418A
3419A
3418B
3419B
3418C
3419C
3418D
3419D
CLK CSA
CLK CSB
CLK CSC
CLK CSD
CLK RWA
CLK RWB
CLK RWC
CLK RWD
2715M
CONTROLLER
IENA
OENA
IENB
OENB
IENC
OENC
IEND
OEND
CLK DBIN 2718M
FROM MEMORY CONTROLLER
CLK DBOUT 2719M
TO MEMORY CONTROLLER

FIG. 34

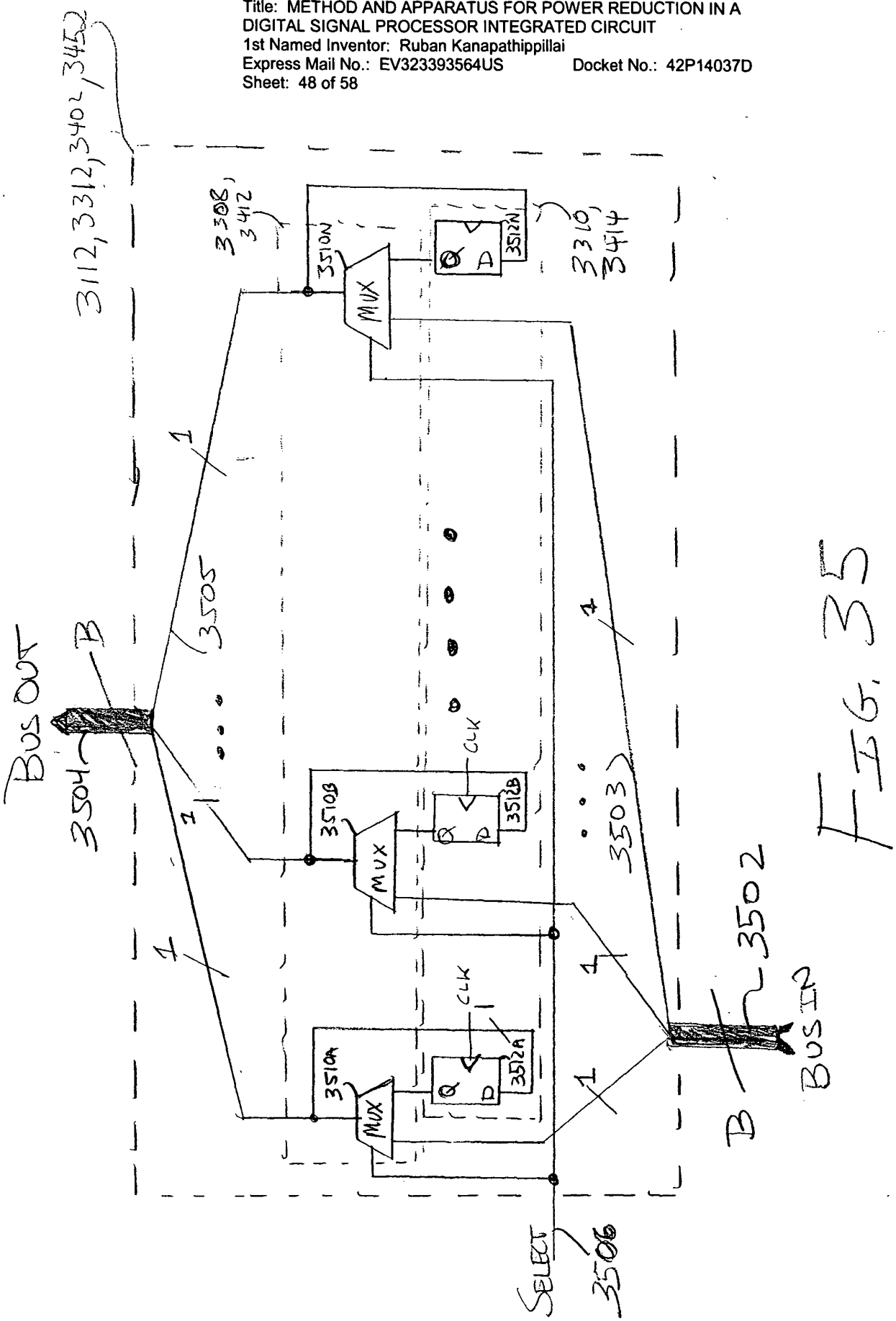

3112, 3312, 3402, 3452
3308, 3412
3310, 3414
3510N
MUX
3512N
D
Q
CLK
3505
1
3504
BUS OUT
B
3510B
3512B
3503
3510A
3512A
3502
BUS IN
SELECT
3506

FIG. 35

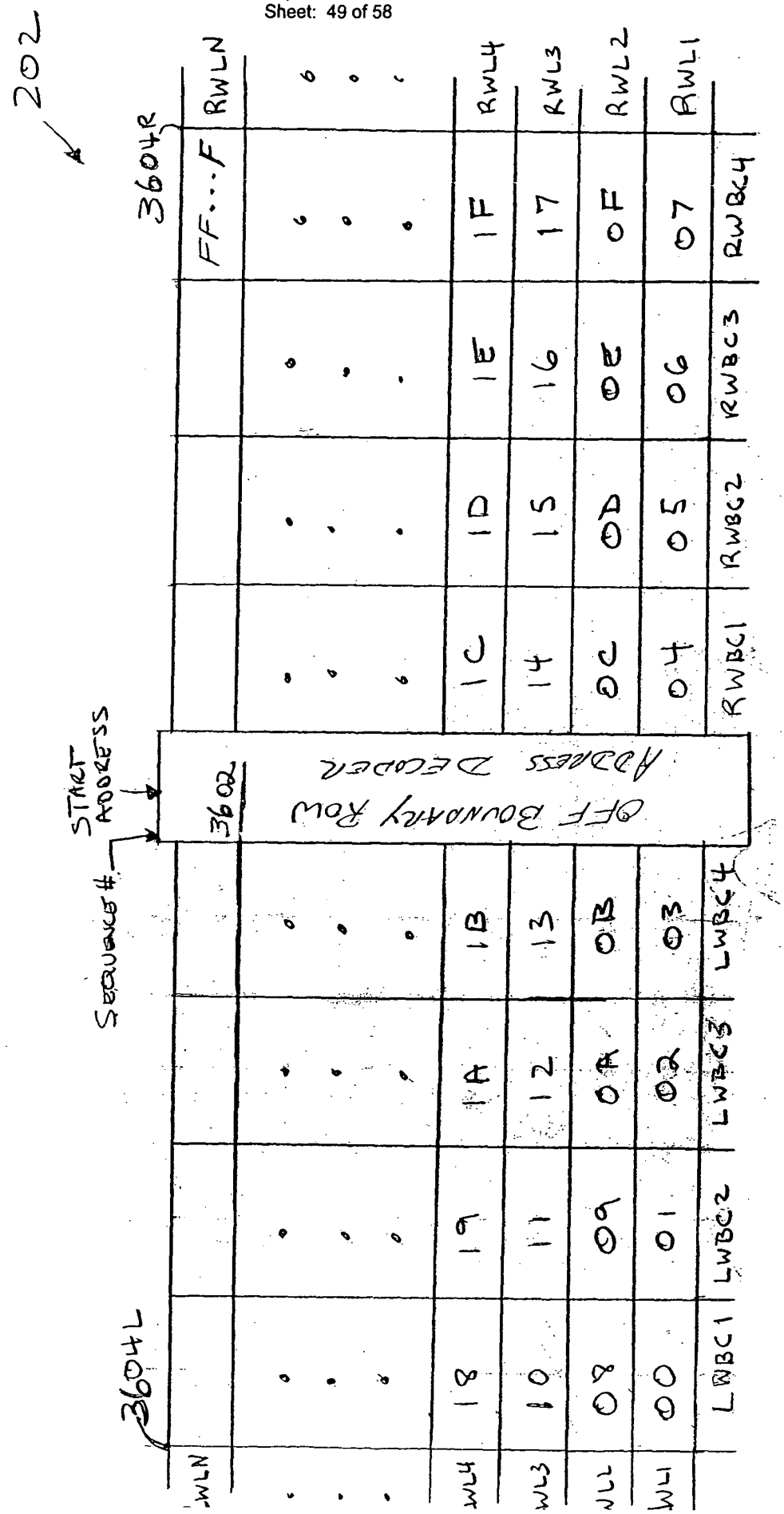

202
3604R
3604L
3602
START ADDRESS
SEQUENCE #
OFF BOUNDARY ROW ADDRESS DECODER
RWLN
RWL4
RWL3
RWL2
RWL1
FF...F
RWBC4
RWBC3
RWBC2
RWBC1
LWBC4
LWBC3
LWBC2
LWBC1
WLN
WL4
WL3
WL2
WL1

FIG. 36A

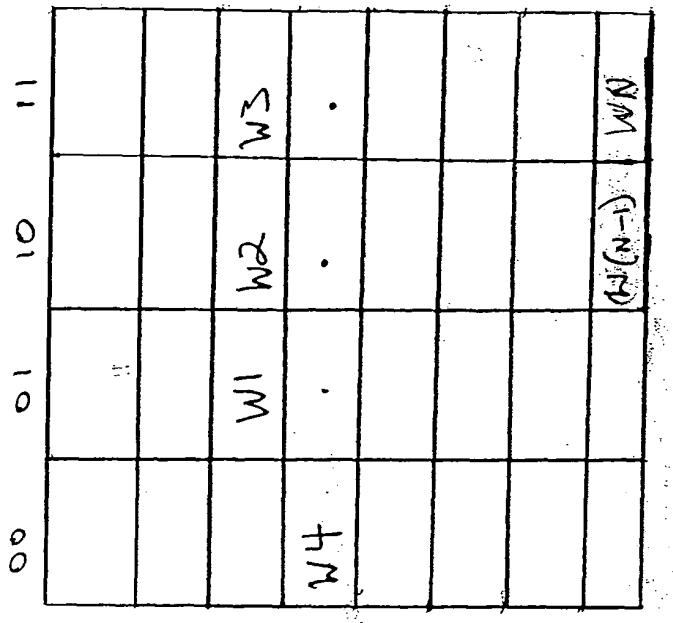


FIG. 36C

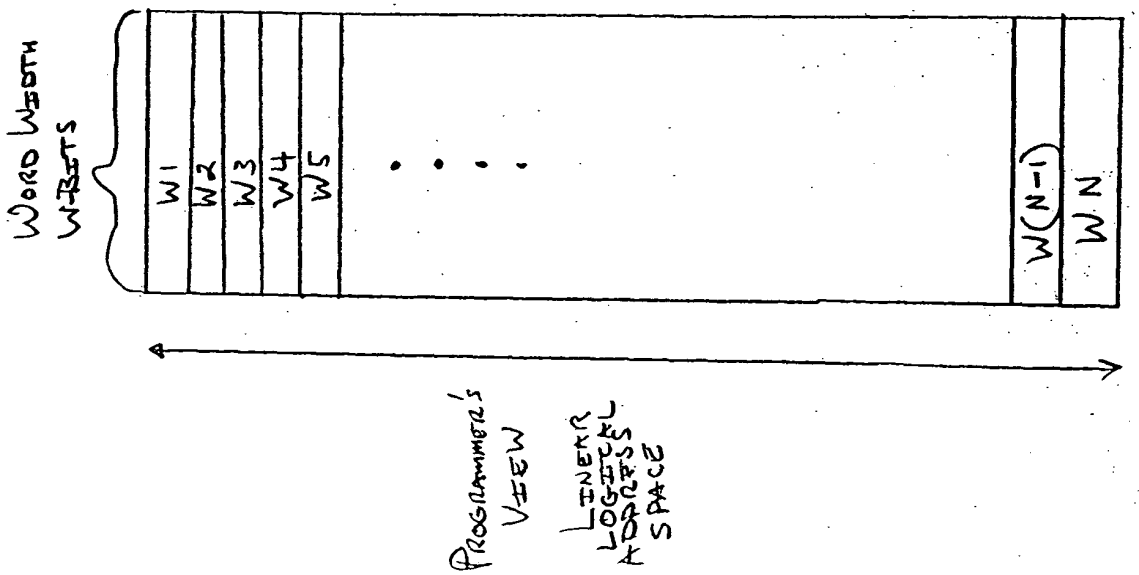


FIG. 36B

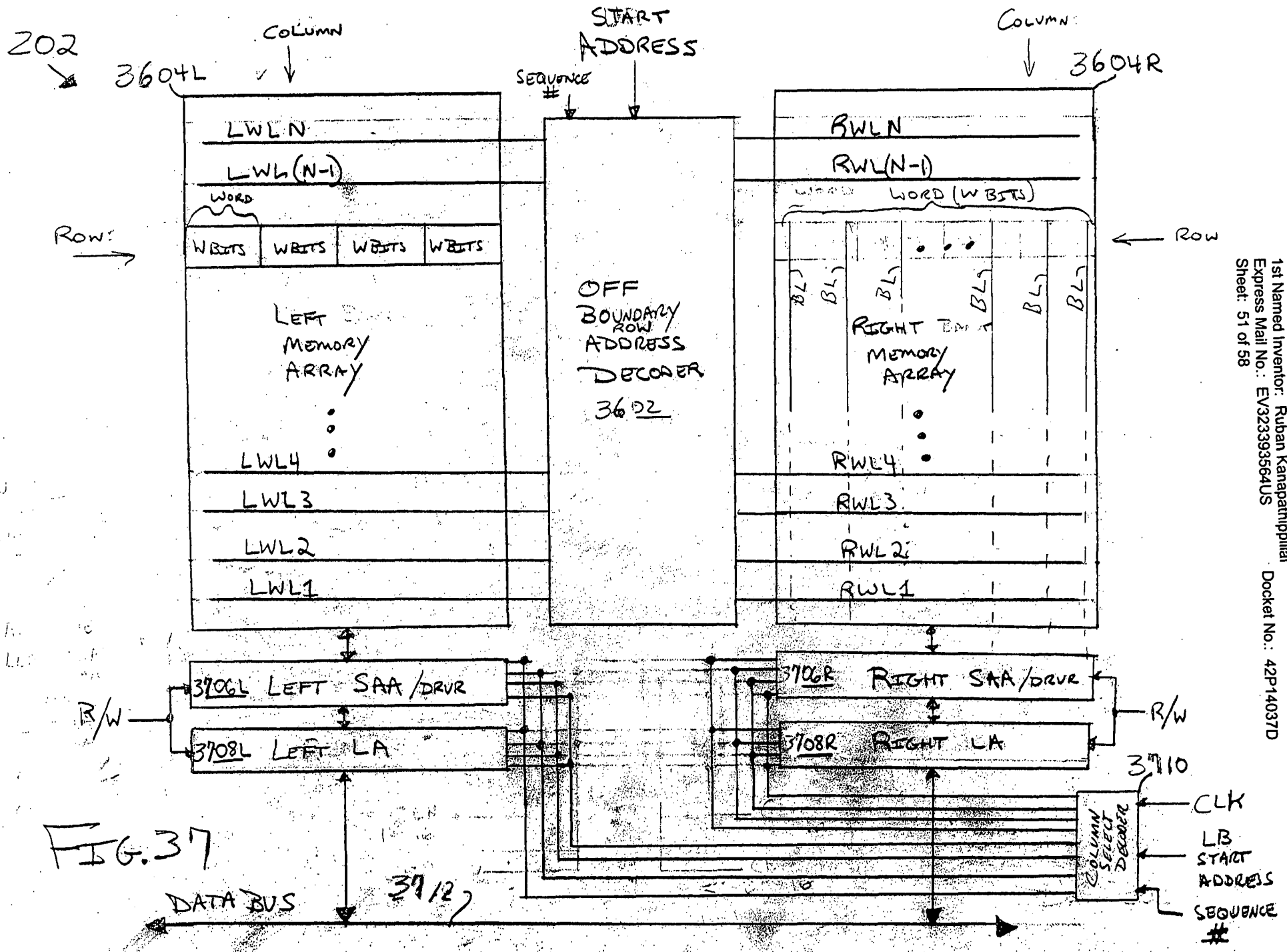


FIG. 37

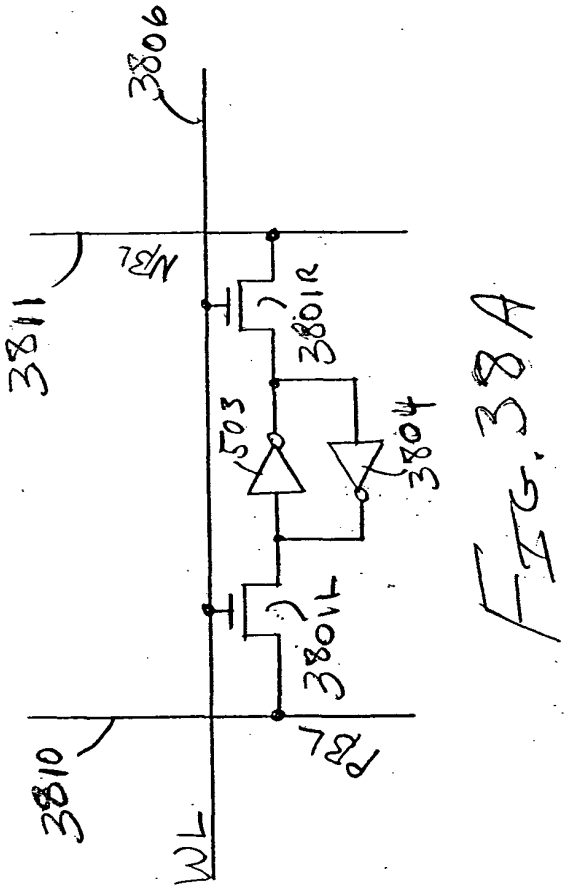


FIG. 38A

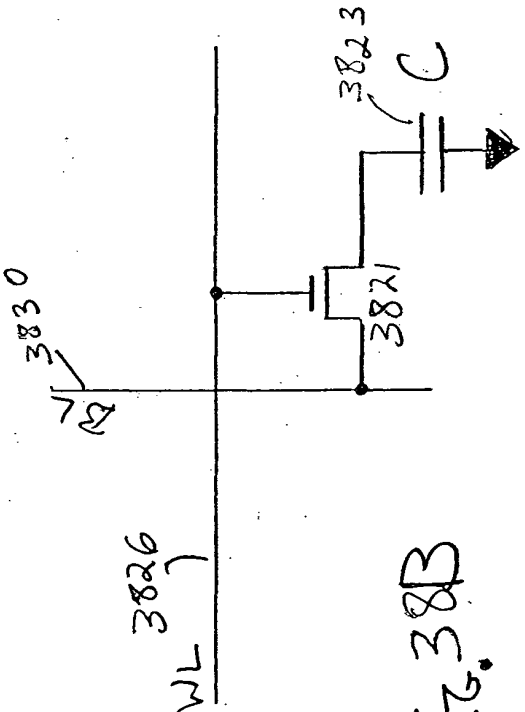


FIG. 38B

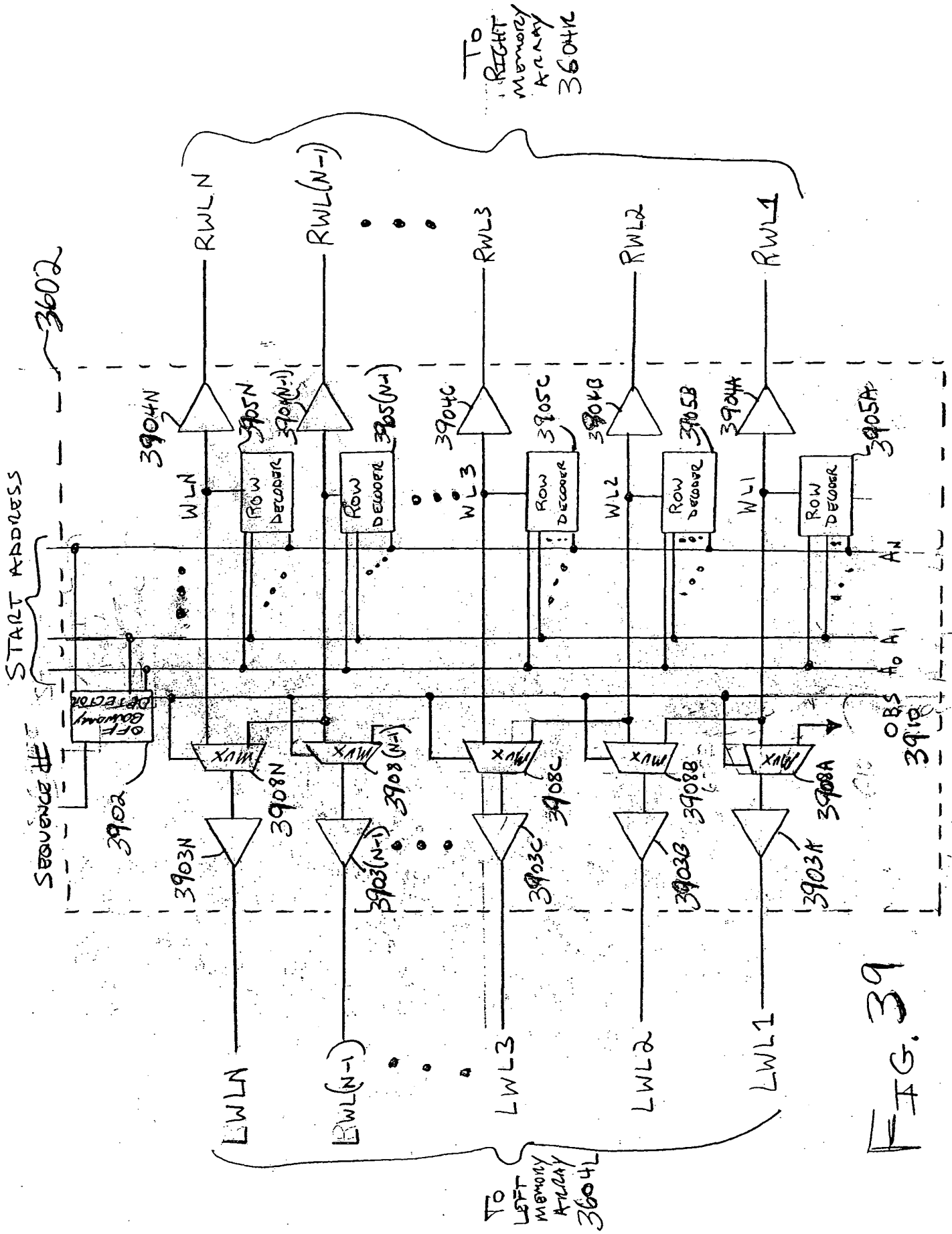

To Right Memory Array 3604R
RWLN
RWL(N-1)
RWL3
RWL2
RWL1
3602
3904N
3905N
3904(N-1)
3905(N-1)
3904C
3905C
3904B
3905B
3904A
3905A
WLN
WL3
WL2
WL1
ROW DECODER
START ADDRESS
SEQUENCE #
OFF BOUNDARY DETECTOR
3902
AN
A1
A0
OBS
3910
MUX
3908N
3908(N-1)
3908C
3908B
3908A
3903N
3903(N-1)
3903C
3903B
3903A
LWLN
LWL(N-1)
LWL3
LWL2
LWL1
To Left Memory Array 3604L

FIG. 39

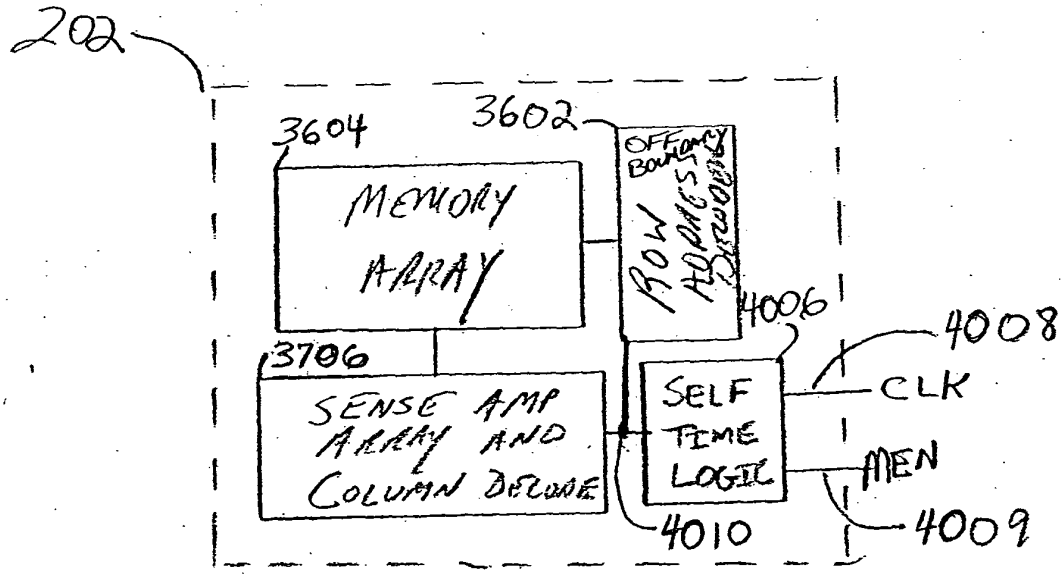

202
3604
3602
MEMORY ARRAY
ROW ADDRESS DECODER
4006
4008
3706
SENSE AMP ARRAY AND COLUMN DECODE
SELF TIME LOGIC
CLK
MEN
4010
4009

FIG. 40

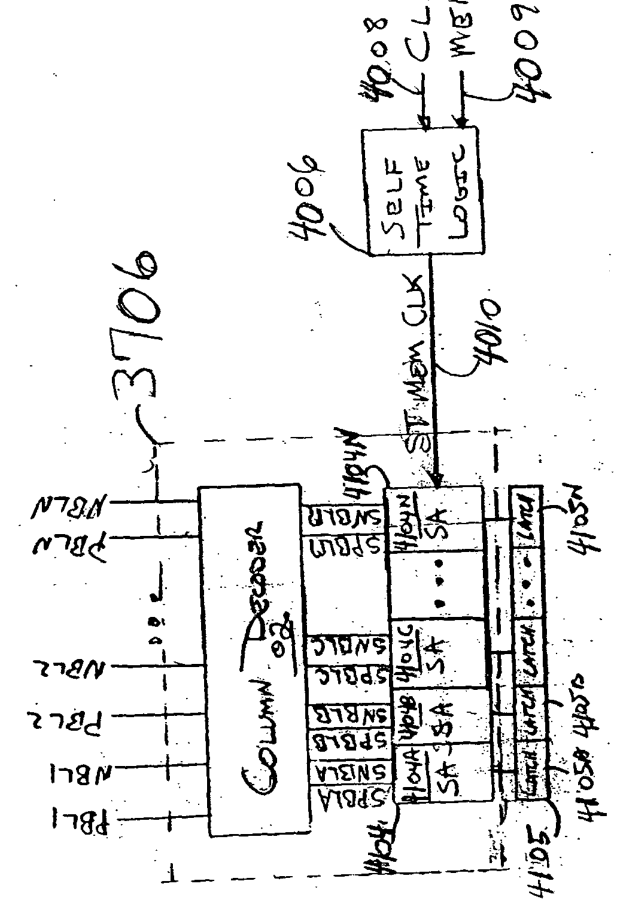


FIG. 41

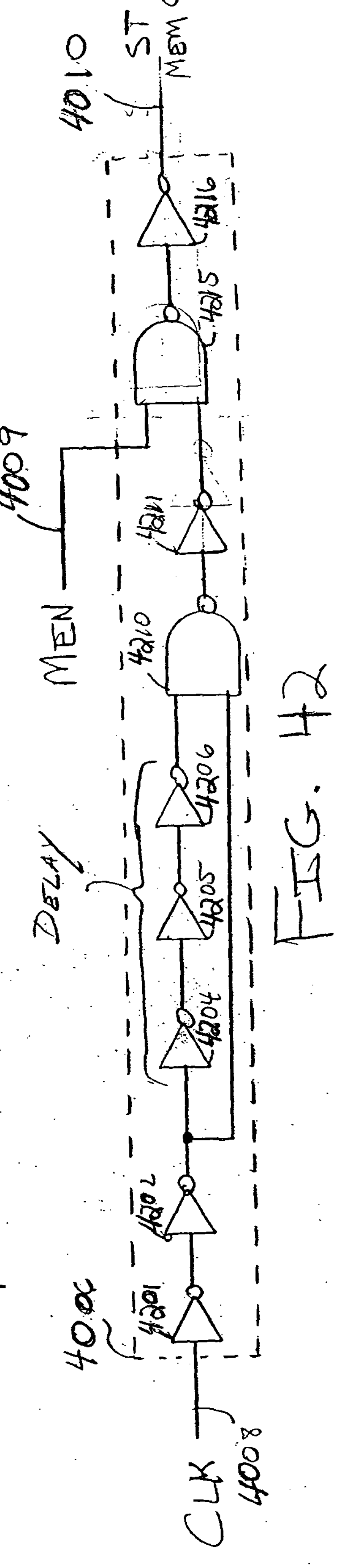


FIG. 42

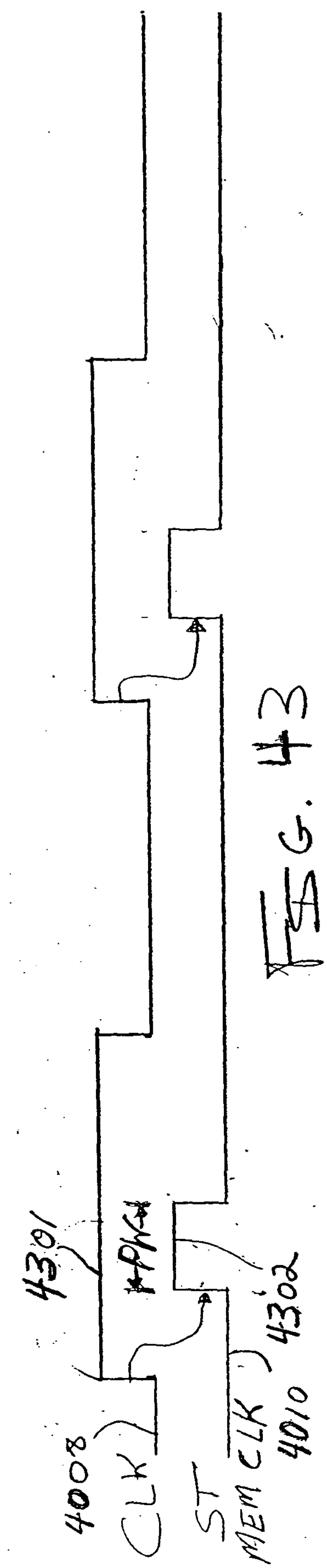


FIG. 43

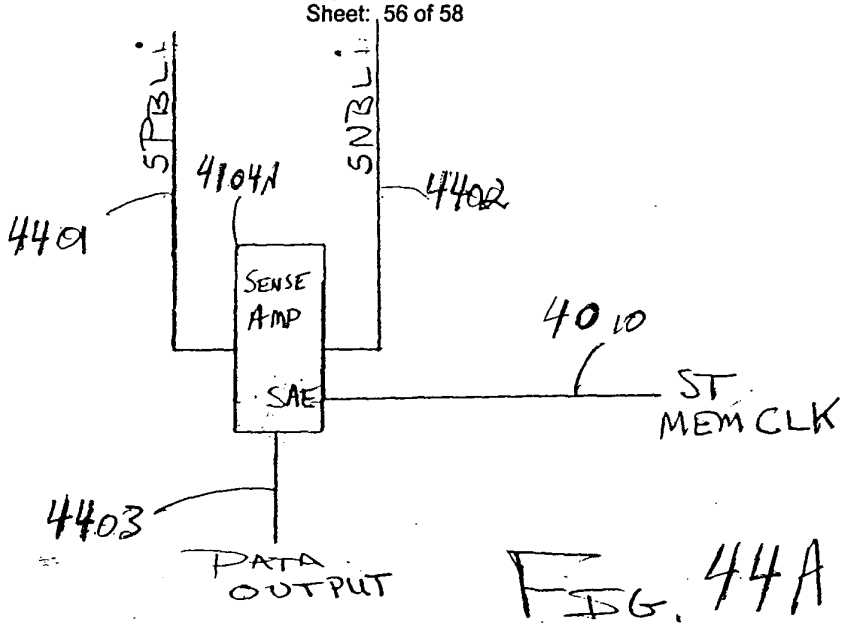


FIG. 44A

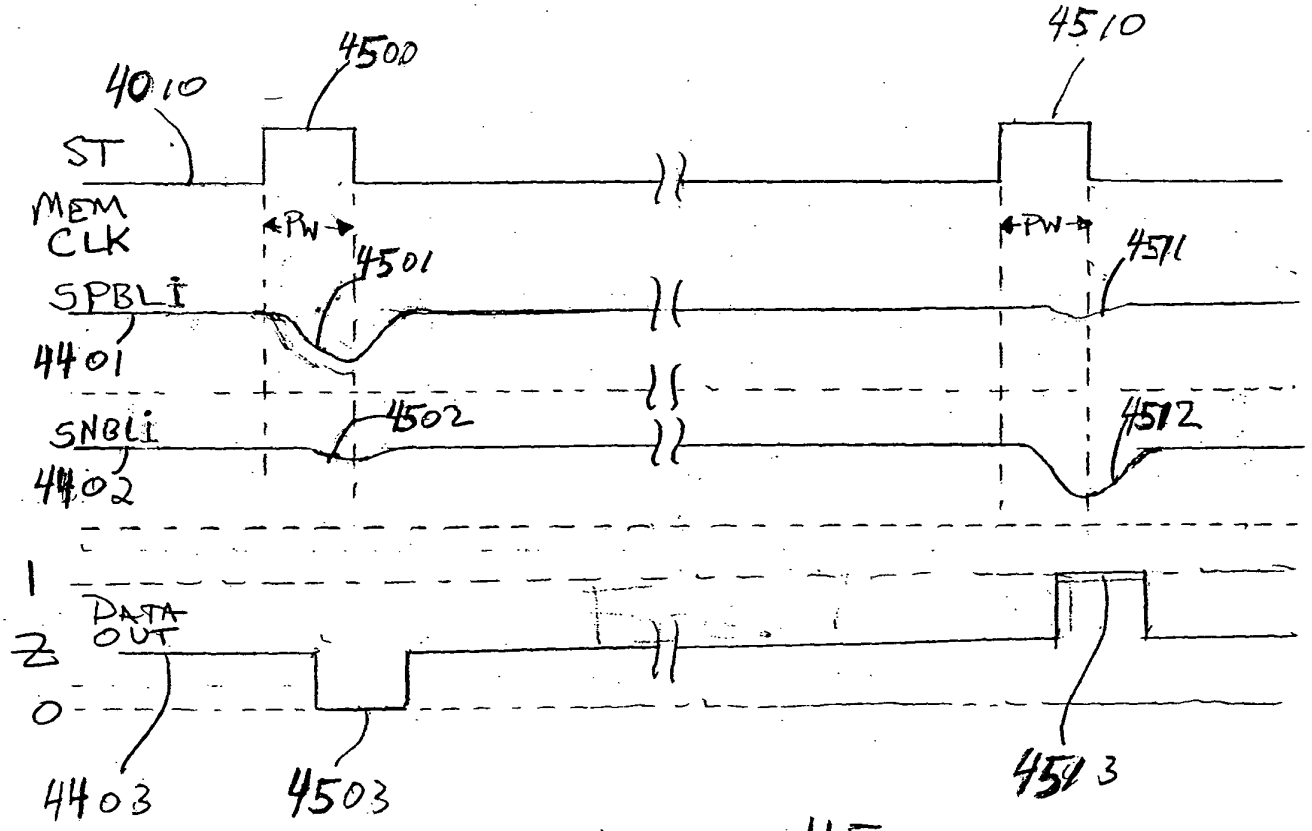


FIG. 45

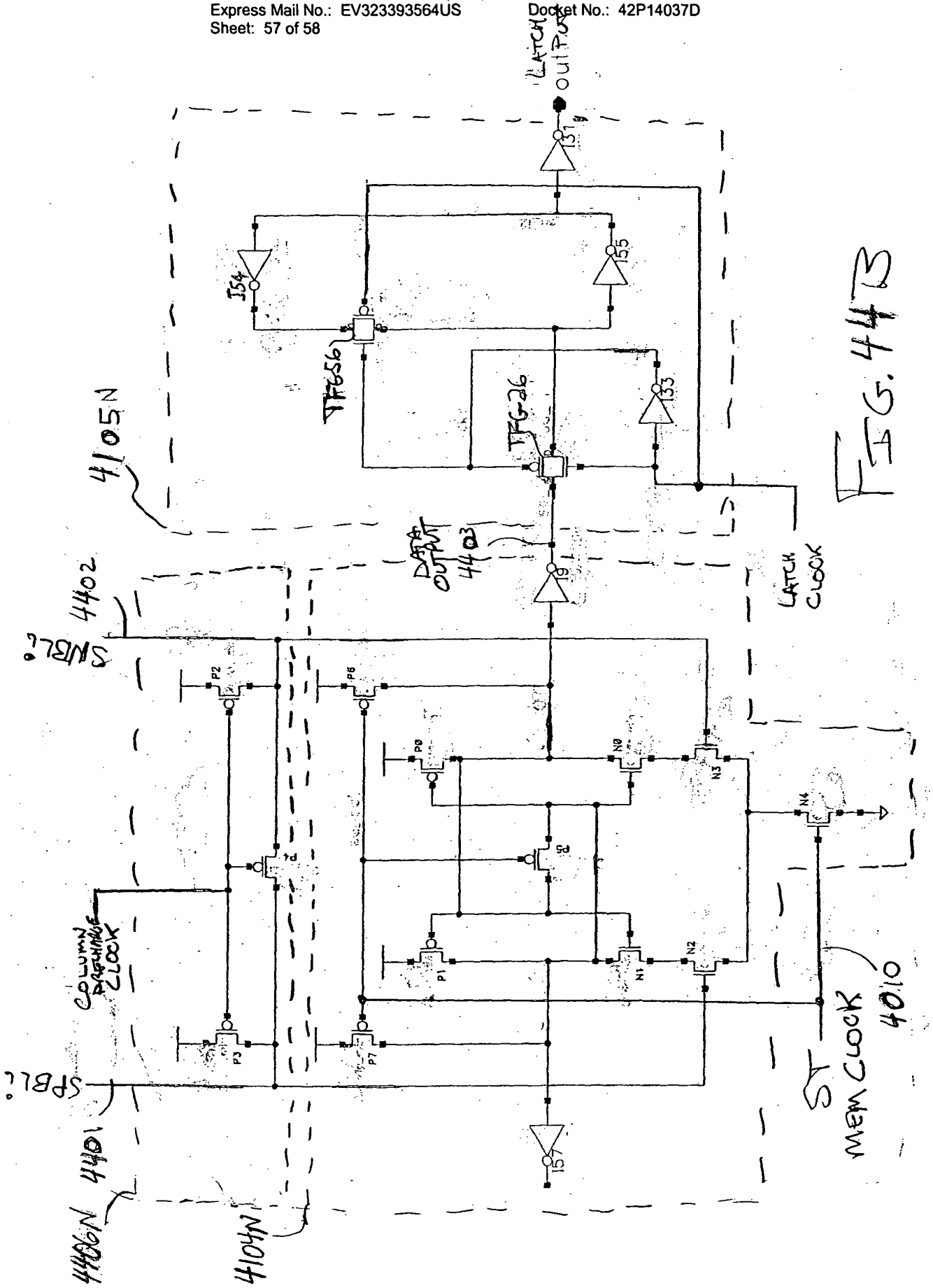

LATCH OUTPUT
151
154
155
TPG56
TPG26
153
4105N
DATA OUTPUT
4403
LATCH CLOCK
4402
SRBL0
P2
P6
P0
N0
N3
N4
P5
P4
COLUMN ADDRESS CLOCK
P1
N1
N2
P3
P7
SRBL1
4401
157
4106N
4104N
SYSTEM CLOCK
4010

FIG. 44B

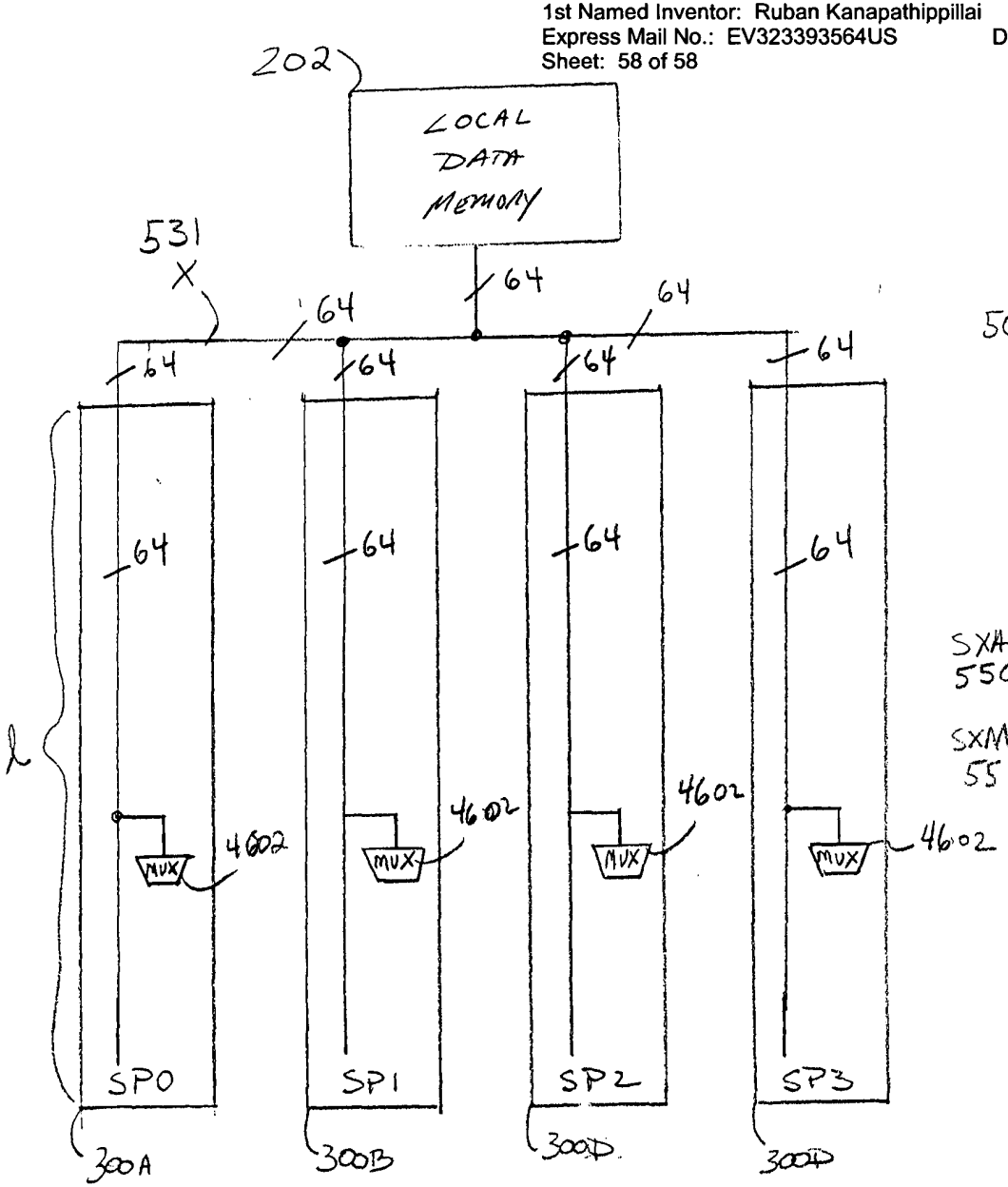


FIG. 46A

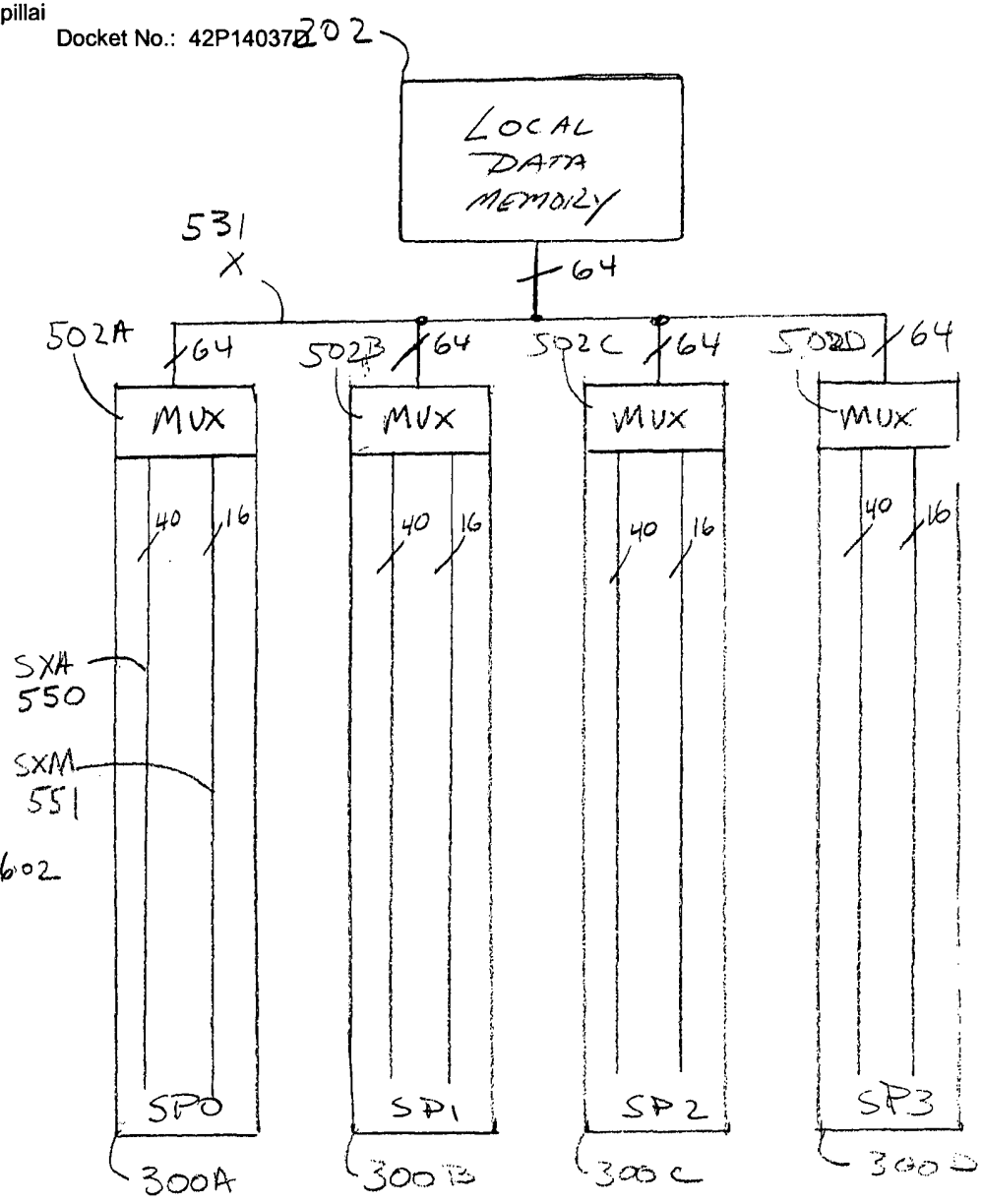


FIG. 46B